PAPER NAME	AUTHOR
Main Project Report ecare.pdf	**K M Abhijith**

WORD COUNT	CHARACTER COUNT
10506 Words	**56466 Characters**
PAGE COUNT	FILE SIZE
66 Pages	**1.7MB**
SUBMISSION DATE	REPORT DATE
Jul 22, 2022 12:16 PM GMT+5:30	**Jul 22, 2022 12:17 PM GMT+5:30**

● 18% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 12% Internet database
- 7% Publications database
- Crossref database

● Excluded from Similarity Report

- Small Matches (Less then 20 words)

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The project E-care includes registration of patients, storing their details into the system, and also book slots to consult doctor. The software stores the details of every patient and the doctor automatically. User can search availability of a doctor.

The E-care can be entered using a username and password. It is accessible either by an administrator or receptionist. The data can be retrieved easily. The interface is very user-friendly.

1.2 PROJECT SPECIFICATION

The purpose of the project entitled as "E-CARE" is to computerize the Front Office Management of Hospital and to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of patient's information, diagnosis details, etc.

The system includes 3 modules. They are:

1.Admin:

- **Login :** Admin need to login**.**
- **Patients:** Admin can view patient's details.
- **Manage department** : Admin can add/manage doctors , departments and Health packages.
- **Reports** : Generate Reports.
    - Appointments for each doctor.
    - Tests done in lab.

Admin can also recover his/her own password.

2. Patient:

- **Login/Register:** patient can login using their id or can register as a new patient
- **Dashboard:** Patients can view the his/her profile, Appointments.
- **Book Appointment** : Patient can book his/her appointment.
- **Find doctor :** View doctors profile .
- **Medical History:** Patients can see his/her own medical history.
- **Lab reports** : View lab reports
- **payment :** Doctor fee payment ,Health package fee payment
- **Health packages**: Book health packages

User can update his/her profile, recover the password.

3.Doctor:

- **Login :** Doctors need to login using their id provided by the admin**.**
- **Dashboard :** Doctor can view his/her own profile and online appointments.
- **Medical History :** Doctor can see patient’s Medical history and recommend medication or further treatment.
- **Search :** Doctor can search patient with the help of patient name and mobile number.
- **Prescription:** Can add prescription.
- **View Schedule:** View teleconsulting appointments.
- **Manage reports** : Doctor can manage patients medical reports

Doctor can update his profile, recover the password.

4. Lab

- **Login :** Lab incharge need to login using valid credentials **.**
- **Dashboard :** Lab incharge can view pending lab schedule prescribed by Doctors.
- **Results :** Lab incharge can Upload lab results to doctors and patients.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is the process of collecting and analyzing data, diagnosing problems, and using the data to recommend system changes. System users and system developers must communicate deeply during this problem-solving process. Any system development process should begin with system analysis or study. The system is meticulously reviewed and evaluated. The system analyst assumes the role of the questioner and digs deeper into the workings of the current system. The input to the system is defined and the system is considered as a whole. Different procedures may be associated with the results of organizations. Understanding the problem, identifying relevant and important variables, evaluating and synthesizing multiple factors, and choosing the best or at least the most acceptable course of action are all part of the analysis. system. This process needs to be studied in depth using a variety of methods, including questionnaires and interviews. To come to a conclusion, the information gathered by these sources must be considered carefully. Understanding how the system works is conclusive. Current system is the name of this system. Now the current system is being carefully reviewed and problem areas are identified. The designer now acts as a problem solver and tries to solve the problems facing the business. Suggestions are made instead of solutions. Then the proposal is analyzed against the current system and the best one is selected. The user has the option to accept or reject the proposal. At the request of the user, the proposal is evaluated and the appropriate modifications are made. As soon as the user is satisfied with the suggestion, this loop is broken. The process of collecting and analyzing data for use in future systematic studies is called preliminary research. Preliminary research is a problem-solving activity that requires close coordination between system users and developers. It does some feasibility studies. These surveys provide a rough estimate of system performance, which can be used to determine what tactics to use for effective system research and analysis.

2.2 EXISTING SYSTEM

Currently, the majority of hospitals employ software where the administrator has complete control over the data. Critical information, including the patient's medical information, is handled by the admin in the majority of the systems currently in use, raising the possibility of its misuse.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Patient information is unsafe.
- The existing system might not be user-friendly to patients with less computer knowledge.
- Difficult to know the availability of doctors.

2.4 PROPOSED SYSTEM

Any hospital can use the E-care to replace their current system. The purpose of the new system is to manage patient data, doctor's appointments, patient information.

In the proposed system, doctors are in charge of managing the crucial patient information. Nobody can access it, not even the administrator.

These services must be delivered effectively and economically with the aim of lowering the time and resources currently needed for similar jobs.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

➢ **Eliminate the chance for critical data misuse:-**

Only doctors have access to the patient's medical records; no other staff members are permitted. Therefore, the patient data will be kept private.

- **Ensure data accuracy: -**

The suggested solution does away with human mistakes made when entering user information during registration.

- **Better service: -**

Hard copy storage won't be a strain for the system. For performing the same activity, we can also save time and resources. The data can be kept for a longer time without losing any information.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

In this phase, the feasibility of the project is evaluated and business proposals are presented along with a very general project plan and some cost estimates. The utility of the proposed system should be considered as part of the system analysis. This ensures that the proposed solution does not burden the company. Understanding the key system requirements is important for feasibility analysis.

Three important considerations in the feasibility study are::

3.1.1 Economical Feasibility

This survey is conducted to determine the financial impact of an organization's system. The company has a limited amount of money to invest in system research and development. Costs should be documented with evidence. As a result, the developed system was below budget. This was made possible by the fact that most technologies were in the public domain. You only need to buy custom items.

3.1.2 Technical Feasibility

This study is conducted to assess the technical requirements of the system or its technical feasibility. The created system should not overwhelm the available technical resources. As a result, our customers are facing great expectations. The system created should have low implementation barriers and require no or few changes.

3.1.3 Behavioral Feasibility

The purpose of this study is to determine how much users will accept the system. This includes the instructions that the user needs to operate the system effectively. The system should not make the user feel threatened. Instead, they should consider it a necessity. The techniques used to inform users and familiarize them with the system are the only factors that influence user acceptance. As the ultimate user of the system, they need to build their confidence

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	-	Intel core i3
RAM	-	4 GB
Hard disk	-	1 TB

3.2.2 Software Specification

Front End	-	HTML, CSS ,BOOTSTRAP,JQUERY
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server-aspect scripting language used for general-motive programming in addition to internet development. More than 244 million web sites and 2.1 million internet servers presently use PHP. The PHP organization now produces the reference implementation of PHP, which turned into first evolved with the aid of using Rasmus Ledorf in 1995. PHP, a recursive acronym that when intended for non-public Home page, now stands for PHP: Hypertext Preprocessor. A internet server`s PHP processor module translates PHP code to supply the very last internet page. In order to deal with data, PHP instructions may be immediately inserted into an HTML supply record in preference to calling an outside record. Due to barriers on using the call PHP, it has additionally evolved to encompass a command-line interface functionality and may be used standalone, which makes it incompatible with the GNU General Public License (GPL). Most internet servers aid the unfastened deployment of PHP, which is likewise to be had as a standalone shell on nearly all systems and running systems.

3.3.2 MySQL

Oracle Oracle Corporation has developed, distributed, and supported MySQL, the most popular open source SQL database management system. More information on the latest MySQL software is available on the MySQL website.

- **MySQL is a database management system.**

The systematic collection of data is called a database. It can be a simple grocery list, a photo gallery, or a large amount of data on a corporate network. A database management system, such as a MySQL server, is required to add, access, and process data stored in computer databases. Computers are good at processing large amounts of data, so a database management system is essential for data processing, whether used as a stand-alone program or as a component of other applications.

- **MySQL databases are relational.**

Instead of putting all your data in one huge warehouse, relational databases store your data in individual tables. Physical files designed for speed include a database structure. The logical model provides a flexible programming environment with objects such as databases, tables, views, rows and columns. One-to-one, one-to-many, unique, mandatory or optional, and "pointers" between different tables are examples of rules that can be built to regulate relationships between different data fields. With a well-designed database, the database enforces these rules so that your application does not experience inconsistent, duplicate, orphaned, outdated, or missing data. MySQL stands for "Structured Query Language" with the prefix SQL. The most common standard language for accessing databases is SQL. Depending on your programming environment, you can explicitly enter SQL (such as generate a report), integrate SQL statements into other language code, or use language-specific APIs that obscure SQL syntax. .. SQL is defined via the ANSI / ISO-SQL standard. Since its inception in 1986, the SQL standard has been repeated several times. In this document, "SQL92" refers to the 1992 standard, "SQL: 1999" refers to the 1999 standard, and "SQL: 2003" refers to the latest version of the standard. SQL standards that exist at any given time are called "SQL standards".

- **MySQL software is open source.**

Being open source, anyone can use and modify the software. Anyone can download the MySQL software for free and use it online. Feel free to inspect the source code and modify it as needed. The GPL (GNU General Public License), used by MySQL software, specifies what you can and cannot do with the software under certain circumstances. If you are concerned about the GPL, or if you need to integrate your MySQL code into a commercial application, you can purchase a commercial licensed version from us. For more information, see MySQL licensing overview.

- **MySQL database server is very fast, reliable, scalable and easy to use.**

If that's what you're looking for, you should give it a try. In addition to other apps, web servers, and other software, the MySQL server runs smoothly on your desktop or laptop and requires little or no maintenance. If you want your entire machine to be dedicated to MySQL, you can change the settings to use all memory, CPU power, and I / O capacity.

- **The MySQL server runs on a client / server or embedded system.**

MySQL database software is a client / server system that includes multiple client programs and libraries, management tools, and a multithreaded SQL server that supports various application programming interfaces (APIs). In addition, it provides a MySQL server as a built-in multithreaded library. You can programmatically incorporate this library into smaller, faster, and easier-to-use stand-alone products.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Any engineered device or product`s improvement technique starts offevolved with layout. A innovative technique is layout. The mystery to an green device is a respectable layout. The technique of the usage of exceptional methodologies and ideas to specify a technique or a device in sufficient element to permit for its bodily implementation is noted as "layout." One manner to explain it's far because the technique of the usage of exceptional methodologies and ideas to specify a device, a technique, or a device in sufficient element to permit for its bodily reality. Regardless of the improvement paradigm this is employed, software program layout bureaucracy the technical middle of the software program engineering technique. The architectural element had to assemble a device or product is advanced thru the device layout. This programme has additionally thru the high-quality viable layout phase, nice tuning all efficiency, performance, and accuracy levels, as withinside the case of any systematic technique. A user-orientated file is transformed right into a file for programmers or database body of workers during the layout phase. The levels of device layout improvement are logical layout and bodily layout

4.2 UML DIAGRAM

A not unusual place language called UML is used to specify, visualize, build, and file the software program device artefacts. The Object Management Group (OMG) became chargeable for growing UML, and a draught of the UML 1.zero definition became offered to the OMG in January 1997.

Unified Modeling Language is called UML. Compared to different famous programming languages like C++, Java, COBOL, etc., UML is unique. A visible language known as UML is used to create software program blueprints. A general-cause visible modelling language for software program device visualization, specification, construction, and documentation is what UML is called. UML isn't always simply used to symbolize software program structures, no matter the truth that that is its maximum not unusual place application. It is likewise used to version structures that aren't software program-based. For instance, the producing facility's technique flow, etc. Although UML isn't always a programming language, equipment may be used to generate code the usage of UML diagrams in a number of languages. The evaluation and layout of objects-orientated structures are at once associated with UML.

UML has been standardized to the factor wherein it's far now an OMG standard. A complete UML diagram that depicts a device is made of all of the factors and relationships. The maximum critical thing of the complete method is the UML diagram's aesthetic impact. It is finished via way of means of the usage of all of the extra components. The following 9 diagrams are a part of UML.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

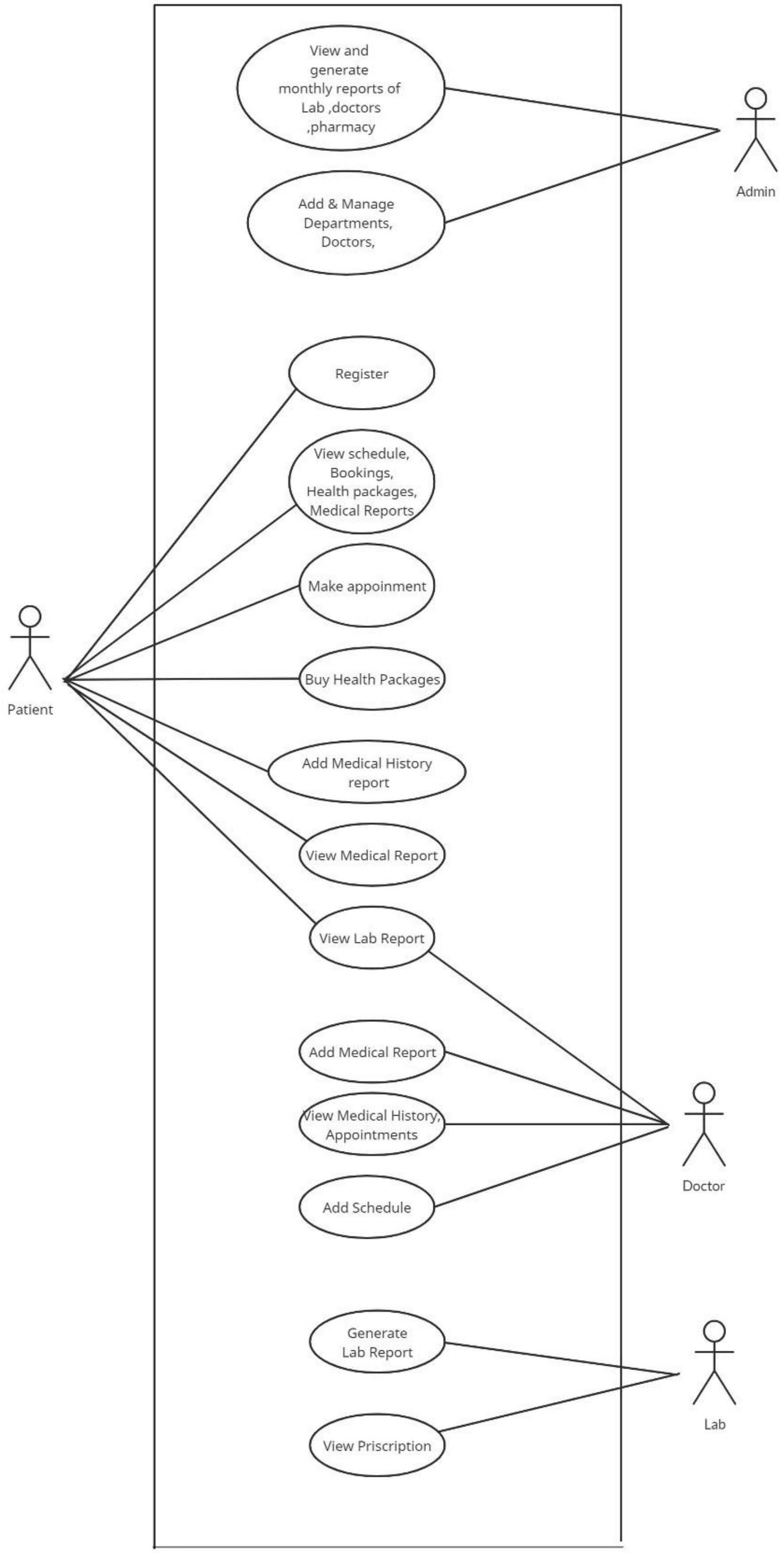
4.2.1 USE CASE DIAGRAM

Use case diagrams are a visual representation of the interactions between system components. The approach for identifying, outlining, and organizing system requirements is called a use case. As used herein, the term "system" refers to anything that is created or operated, such as a website for the sale of shipping products and services. UML (Unified Modeling Language), a standard language for modeling real-world objects and systems, uses use case diagrams. Planning overall requirements, validating hardware designs, testing and debugging software products under development, creating online help references, or performing customer support-oriented tasks are just a few examples of system goals. For example, product sales context use cases may include customer service, item orders, catalog updates, and payment processing. There are four elements in the use case diagram.

- Actors, often the people involved in the system, are defined according to their role. The boundaries that the target system places in relation to its environment.

- Use cases that represent relationships between actors and the exact role they play in and around the system.

Fig 1 : Use case diagram for E-care

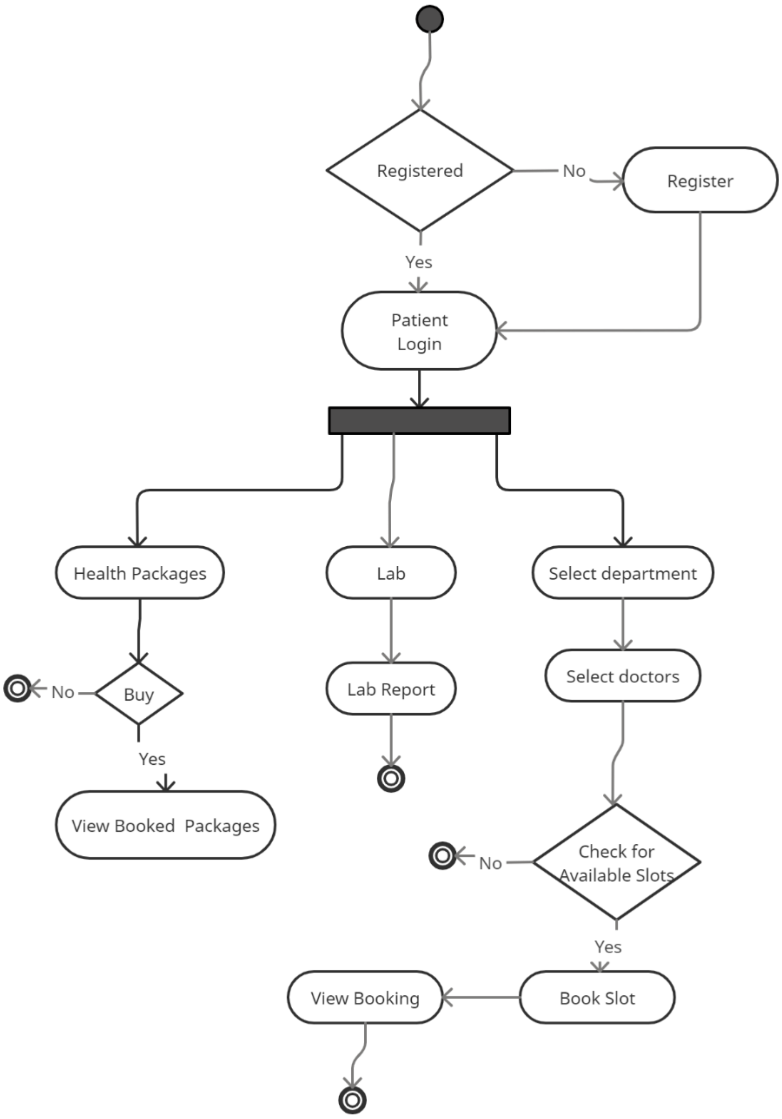

View and generate monthly reports of Lab ,doctors ,pharmacy
Add & Manage Departments, Doctors,
Admin
Register
View schedule, Bookings, Health packages, Medical Reports
Make appoinment
Buy Health Packages
Add Medical History report
View Medical Report
View Lab Report
Patient
Add Medical Report
View Medical History, Appointments
Add Schedule
Doctor
Generate Lab Report
View Priscription
Lab

4.2.2 ACTIVITY DIAGRAM

An activity diagram is essentially a flowchart that shows how one activity leads to another. The action might be referred to as a system operation.

One operation leads to the next in the control flow. This flow may be parallel, contemporaneous, or branched. Activity diagrams use many features, such as fork, join, etc., to cope with all types of flow control.

Fig 2 : Activity diagram for E-care

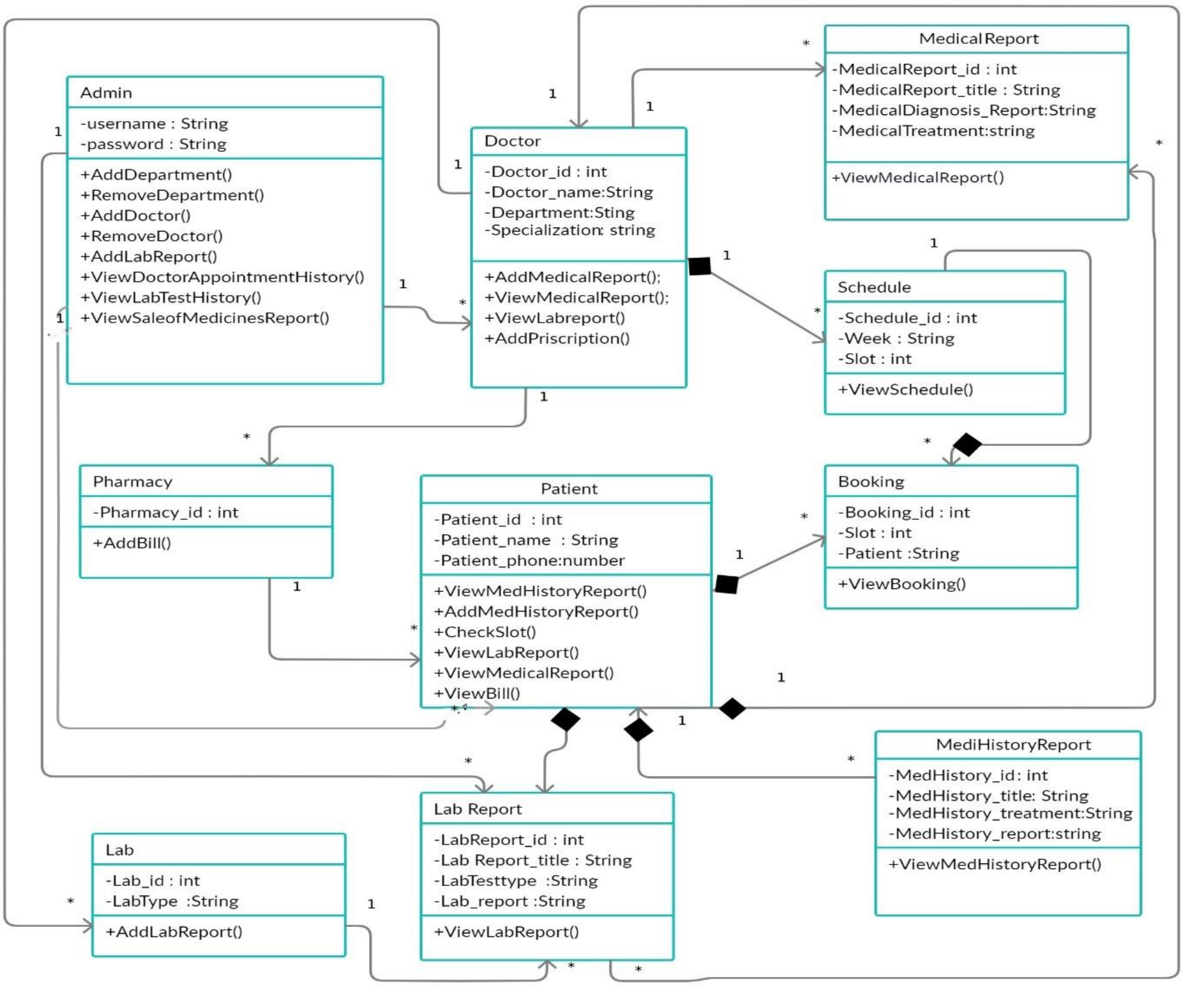


4.2.3 CLASS DIAGRAM

Static diagrams include class diagrams. It represents the application's static view. Class diagrams are used to create executable code for software applications as well as for visualising, explaining, and documenting various elements of systems.

The characteristics and functions of a class are described in a class diagram, along with the restrictions placed on the system. Because they are the only UML diagrams that can be directly translated with object-oriented languages, class diagrams are extensively utilised in the modelling of object oriented systems.

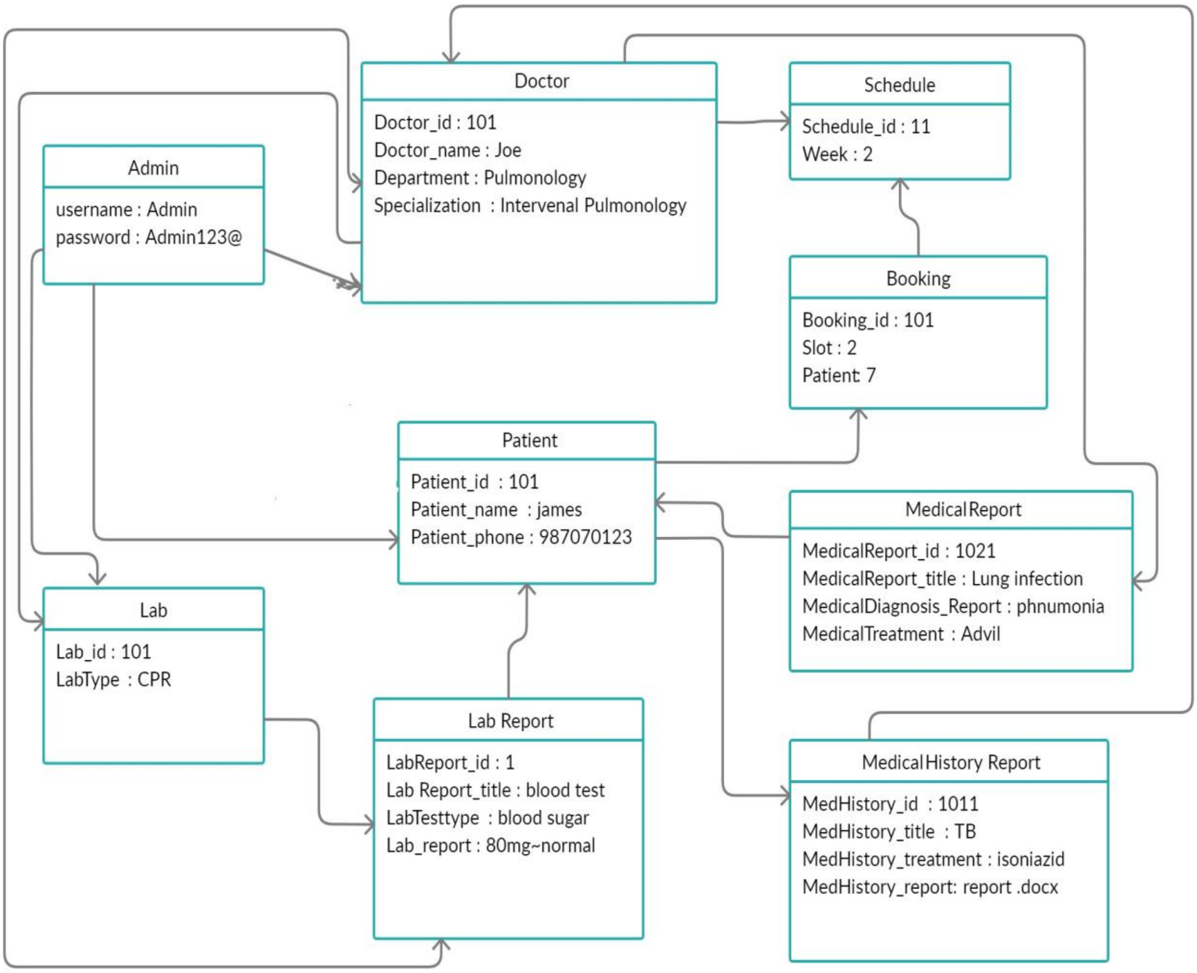
Fig 3 : Class diagram for E-care



4.2.4 OBJECT DIAGRAM

An instance of a class diagram is represented by an object diagram. Class and object diagrams both use the same fundamental ideas. The static view of a system is also represented by object diagrams, but this static view represents a momentary snapshot of the system.

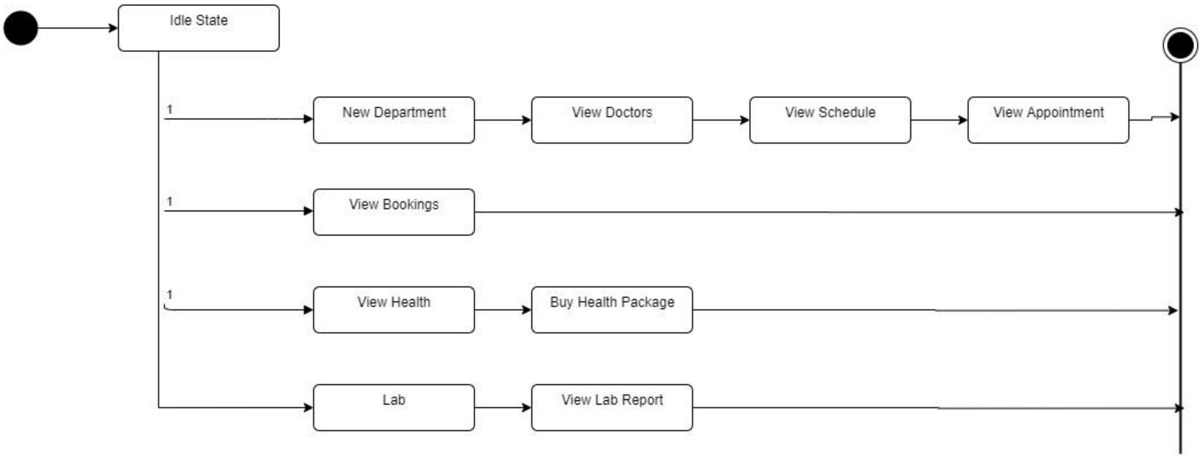
Fig 4 : Object diagram for E-care



4..2.5 STATECHART DIAGRAM

One of the five UML diagrams used to depict a system's dynamic nature is the statechart diagram. Throughout an object's existence, they define several states, and these states are altered by events. The reactive systems can be modelled with statechart diagrams. A system that reacts to internal or external events is known as a reactive system.

Fig 5 : Statechart diagram for E-care



4.2.6 SEQUENCE DIAGRAM

A sequence diagram essentially shows how things interact with one another sequentially, or the order in which these interactions occur. A sequence diagram can also be referred to as event diagrams or event scenarios. Sequence diagrams show the actions taken by the components of a system in chronological order. Businesspeople and software engineers frequently use these diagrams to record and comprehend the requirements for new and current systems.

Sequence Diagram Notations –

i. **Actors** – In a UML diagram, an actor represents a particular kind of role in which it communicates with the system's objects. An actor is always beyond the purview of the system that we want to use the UML diagram to represent. We employ actors to portray a variety of roles, including those of human users and other outside subjects. In a UML diagram, an actor is represented using a stick person notation. In a sequence diagram, there might be several actors.

ii. **Lifelines** – A named piece that shows a specific participant in a sequence diagram is called a lifeline. In essence, a lifeline represents each incident in a sequence diagram. The top of a sequence diagram is where the lifeline pieces are placed.

iii. **Messages** – Using messages, communication between objects is demonstrated. The messages are displayed on the lifeline in chronological sequence. Arrows are how messages are represented. A sequence diagram's main components are lifelines and messages.

Messages can be broadly classified into the following categories:

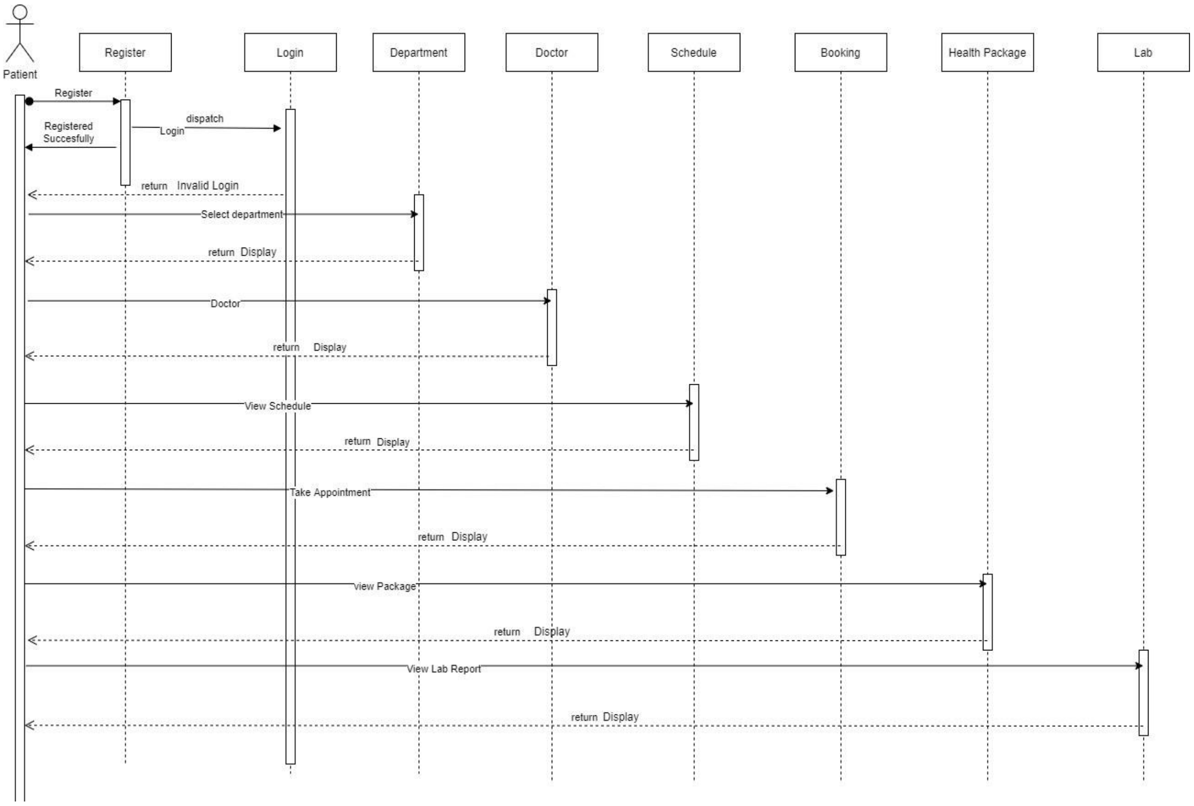
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message

iv. **Guards** – In the UML, we utilise guards to model circumstances. When we need to limit the flow of messages under the guise of a condition being met, we use them. Software engineers rely on guards to inform them of the limitations imposed by a system or specific process.

Uses of sequence diagrams –

- They are also used to display specifics of UML use case diagrams, as well as to describe and depict the logic underlying complex functions, operations, or procedures.
- Used to comprehend the precise operation of present or upcoming systems.
- Visualize the flow of information between different system elements or objects.

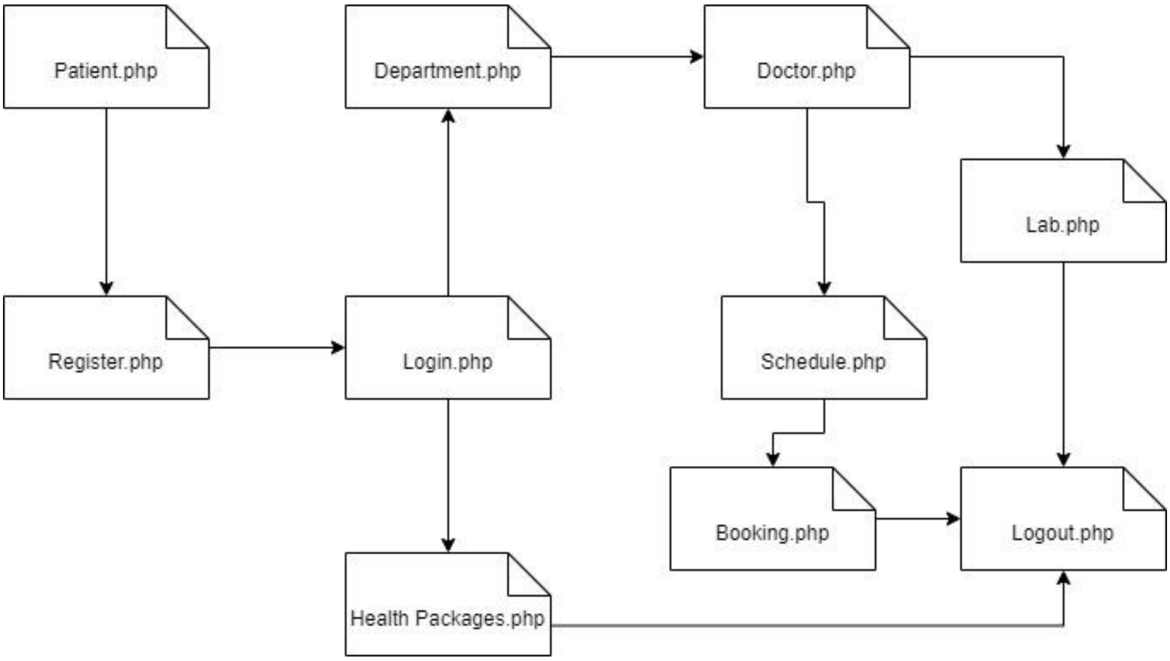
Fig 6 : Sequence diagram for E-care



4.2.7. COMPONENT DIAGRAM

A component diagram, often called a UML component diagram, shows how the physical parts of a system are wired up and organised. Component diagrams are frequently used to model implementation details and confirm that all necessary system functions have been accounted for.

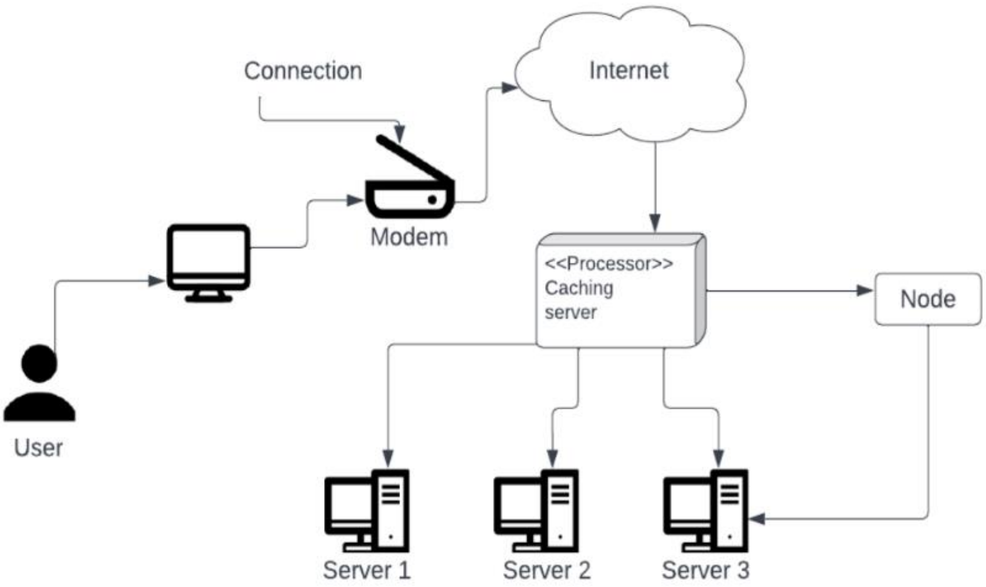
Fig 7 : Component diagram for E-care



4.2.8. DEPLOYMENT DIAGRAM

An execution architecture of a system, containing nodes like hardware or software execution environments, and the middleware linking them, is shown in a deployment diagram, a form of UML diagram. Typically, deployment diagrams are used to represent the actual hardware and software of a system.

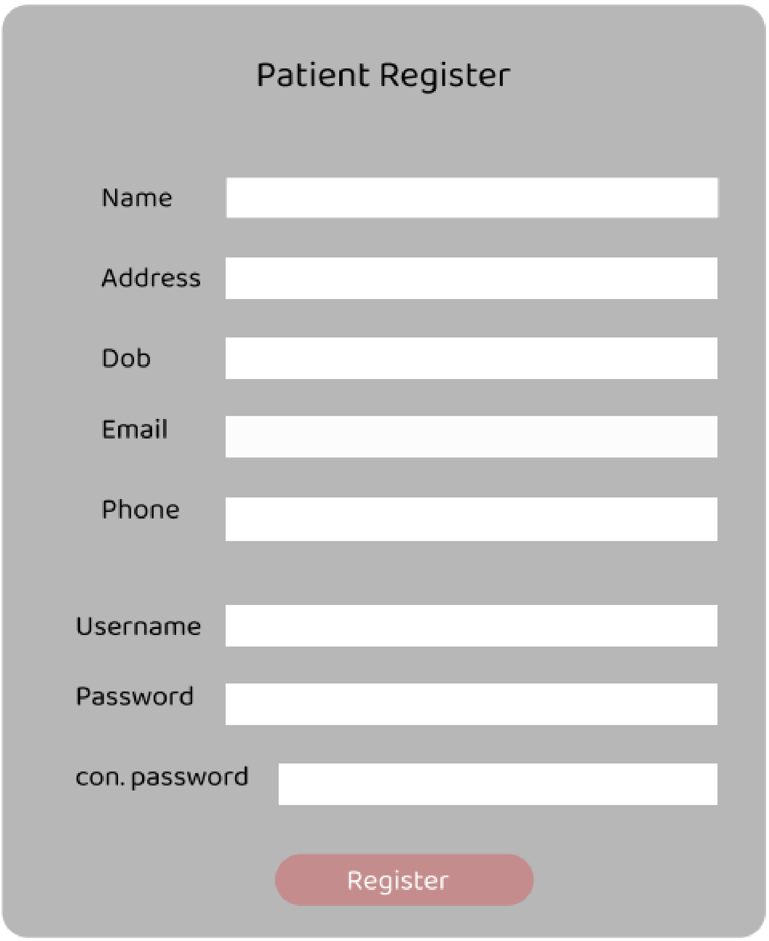
Fig 78: Deployment diagram for E-care



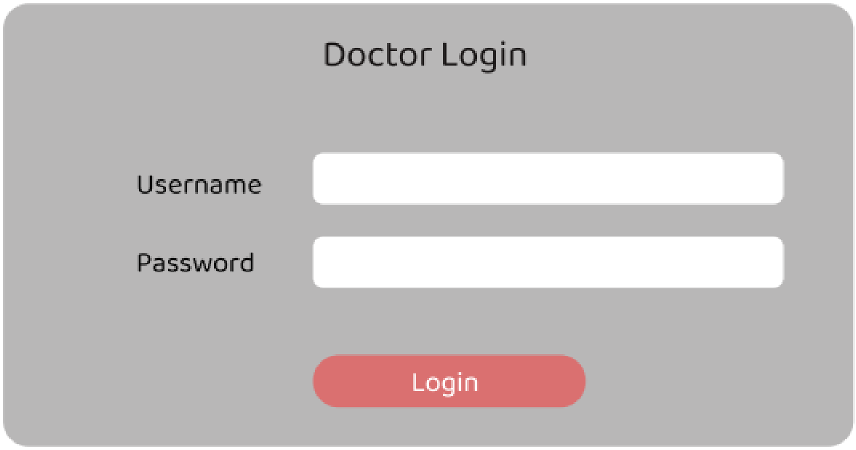
4.3 USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

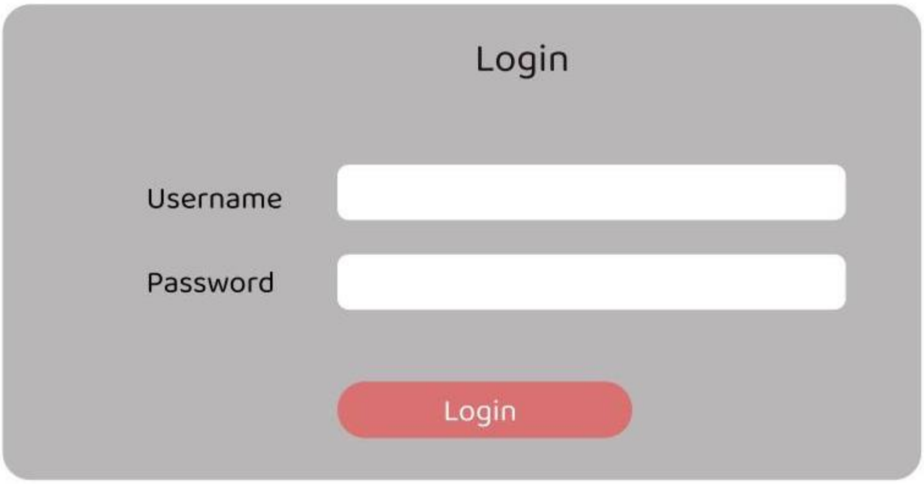
Form Name : Patient Registration



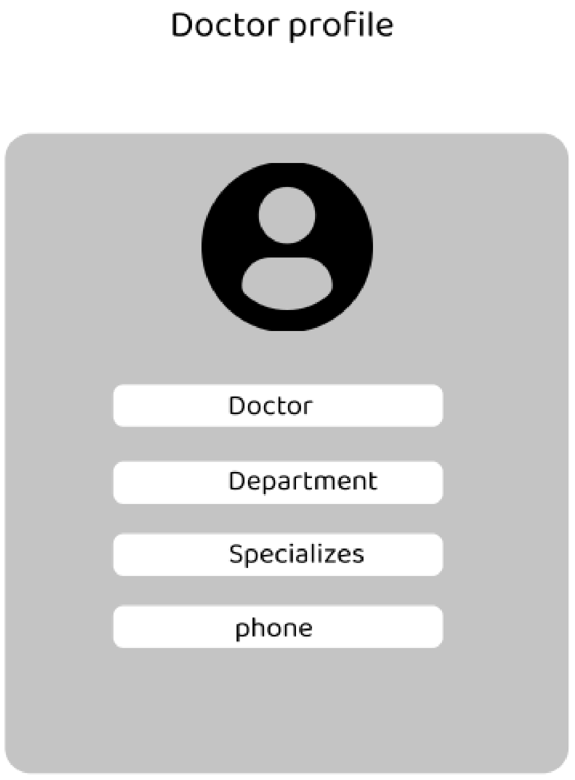
Form Name : Doctor login



Form Name : Patient Login

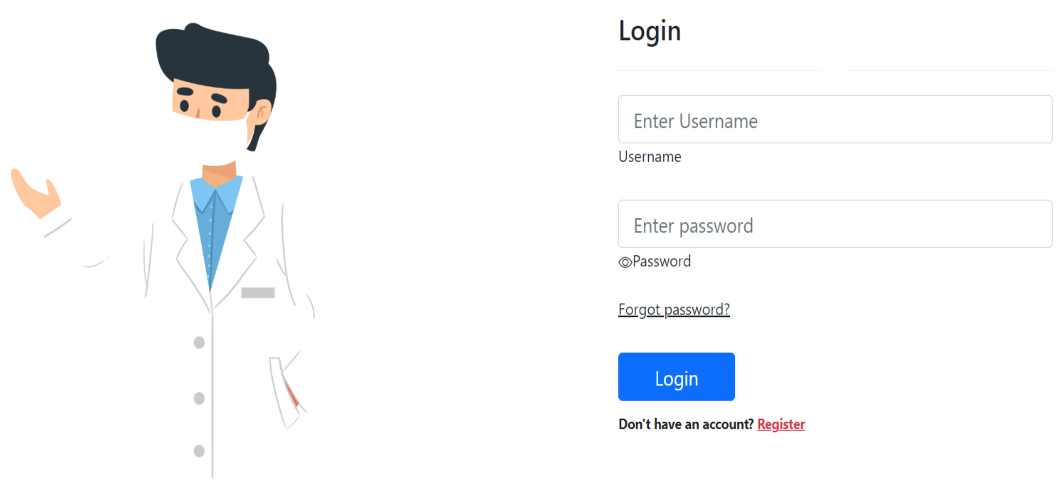


Form Name : Patient viewing doctors profile



4.3.2 OUTPUT DESIGN

User Login



Login

Enter Username

Username

Enter password

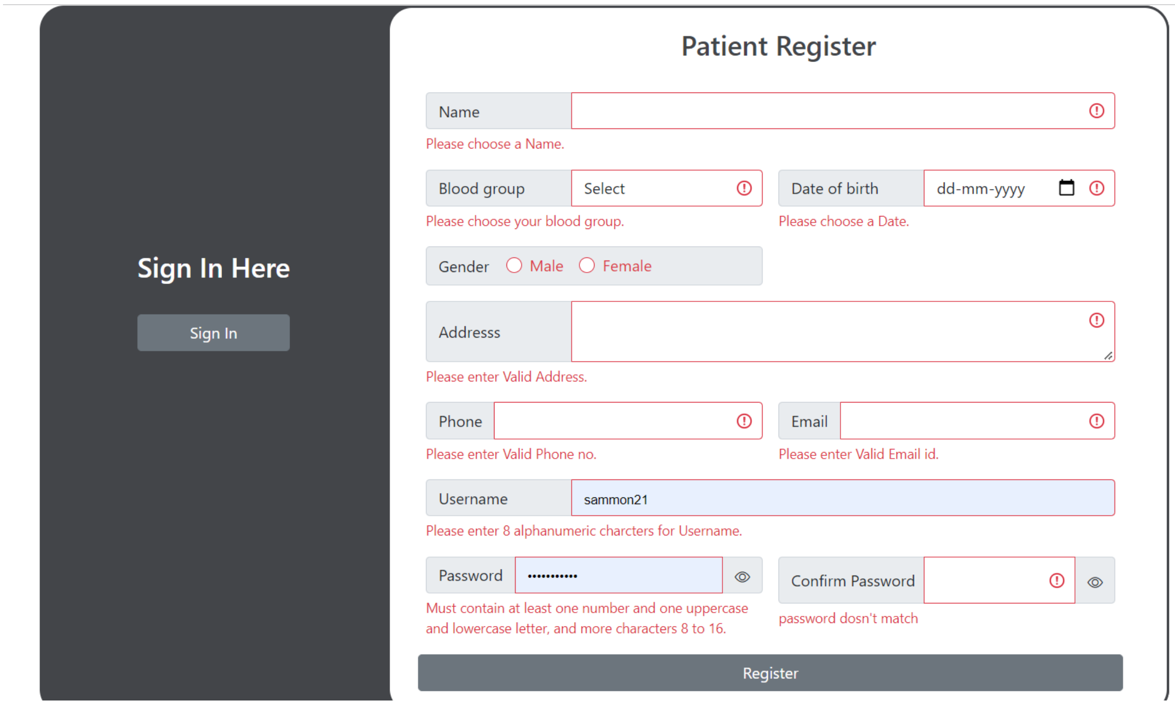
Password

Forgot password?

Login

Don't have an account? Register

Patient Registration



Sign In Here

Sign In

Patient Register

Name

Please choose a Name.

Blood group Select

Please choose your blood group.

Date of birth dd-mm-yyyy

Please choose a Date.

Gender ○ Male ○ Female

Addresss

Please enter Valid Address.

Phone

Please enter Valid Phone no.

Email

Please enter Valid Email id.

Username sammon21

Please enter 8 alphanumeric charcters for Username.

Password ••••••••••••

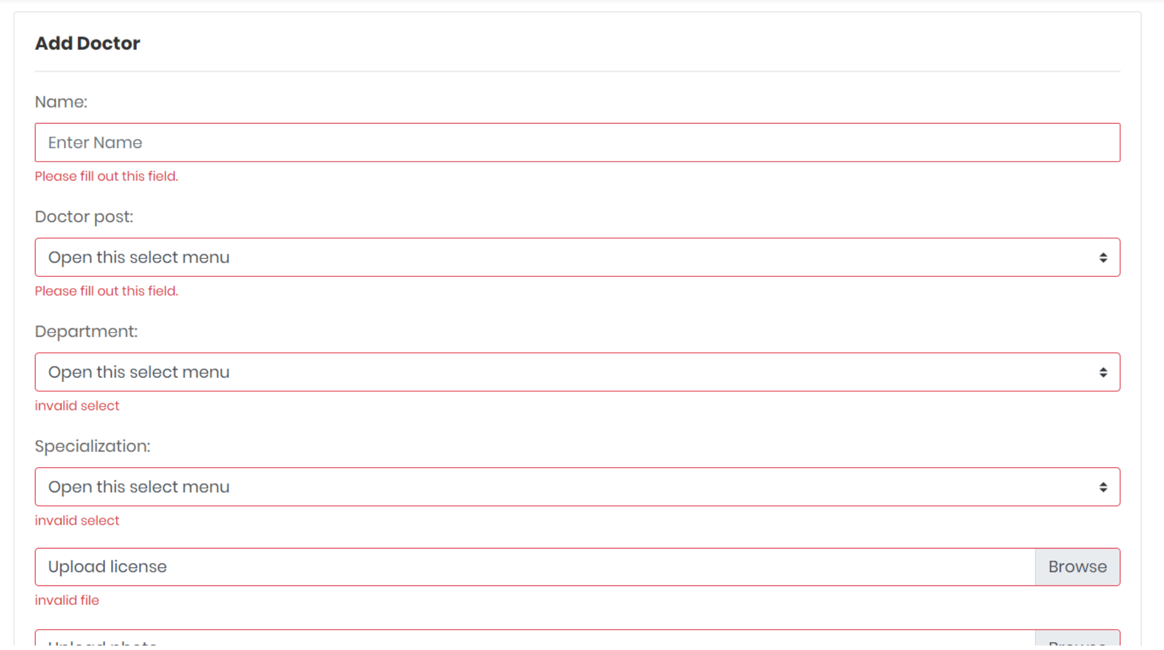
Must contain at least one number and one uppercase and lowercase letter, and more characters 8 to 16.

Confirm Password

password dosn't match

Register

Add Doctor



Add Doctor

Name:

Enter Name

Please fill out this field.

Doctor post:

Open this select menu

Please fill out this field.

Department:

Open this select menu

invalid select

Specialization:

Open this select menu

invalid select

Upload license

invalid file

4.4 DATABASE DESIGN

A database is a structured system with the capacity to store information and allows users to retrieve stored information quickly and effectively. Any database's primary goal is its data, which demands protection.

There are two stages to the database design process. The user needs are obtained in the first step, and a database is created to as clearly as possible meet these criteria. This process, known as information level design, is carried out independently of all DBMSs.

The design for the specific DBMS that will be used to construct the system in issue is converted from an information level design to a design in the second stage. Physical Level Design is the stage where the characteristics of the particular DBMS that will be used are discussed. Parallel to the system design is a database design. The database's data arrangement aims to accomplish the two main goals listed below.

- Data Integrity
- Data independence

4.4.1 Relational Database Management System (RDBMS)

In a relational model, the database is shown as a set of relations. Each relation resembles a file or table of records with values. A row is referred to as a tuple, a column heading is referred to as an attribute, and the table is referred to as a relation in formal relational model language. A relational database is made up of a number of tables, each with its own name. In a story, each row represents a group of associated values.

Relations, Domains & Attributes

A relation is a table. Tuples are the units of a table's rows. An ordered group of n elements is a tuple. Attributes are referred to as columns. Every table in the database has relationships already established between them. This guarantees the integrity of both referential and entity relationships. A group of atomic values make up a domain D. Choosing a data type from which the domain's data values are derived is a typical way to define a domain. To make it easier to understand the values of the domain, it is also helpful to give it a name. Each value in a relation is atomic and cannot be broken down.

Relationships

- Key is used to create table relationships. Primary Key and Foreign Key are the two principal keys that are most crucial. With the use of these keys, relationships for entity integrity and referential integrity can be created.
- Entity Integrity forbids the use of null values for any Primary Key.
- No Primary Key may contain null values, according to Referential Integrity.
- Referential Integrity: A Primary Key value in the same domain must correspond to each unique Foreign Key value. Super Key and Candidate Keys are additional keys.

4.4.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. The formal process of normalising data structures in a way that reduces duplication and fosters integrity. Using the normalisation technique, superfluous fields are removed and a huge table is divided into several smaller ones. Anomalies in insertion, deletion, and updating are also prevented by using it. Keys and relationships are two notions used in the standard form of data modelling. A row in a table is uniquely identified by a key. Primary keys and foreign keys are the two different kinds of keys. A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalised. As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

According to the First Normal Form, each attribute in a tuple must have a single value from the attribute's domain and its domain must only include atomic values. The 1NF forbids "relations within relations" or "relations as attribute values within tuples," in other words. By 1NF, only single atomic or indivisible values are allowed for attribute values. Putting the data into First Normal Form is the first step. Data that is of a similar type in each table can be moved into separate tables to solve this problem. According to the project's requirements, a Primary Key or Foreign Key is assigned to each table. For each nested or non-atomic attribute, we create additional relations in this. For each nested relation or non-atomic attribute, new relations are formed in this process. This got rid of data groups that were repeated. If a relation solely meets the constraints that include the primary key, it is said to be in first normal form.

Second Normal Form

No non-key attribute should be functionally dependent on a portion of the main key for relations where the primary key has several attributes, according to Second Normal Form. This involves breaking down each partial key into its dependent characteristics and setting up a new relation for each one. Keep the original primary key and any properties that are entirely dependent on it in your database. This procedure aids in removing data that depends only on a small portion of the key. If and only if a relation satisfies all the requirements for first normal form for the primary key and all of the non-primary key qualities of the relation are completely dependent on the primary key alone, then that relation is said to be in second normal form.

Third Normal Form

A non-key attribute of a Relation should not be functionally determined by another non-key property or by a collection of non-key attributes, according to the Third Normal Form. In other words, the primary key shouldn't be transitively dependent on anything. In this, we break down the relation into its component parts and build up the non-key qualities that functionally determine the other non-key attributes. To eliminate anything that does not totally dependant on the Primary Key, this step is conducted. A relation is only considered to be in third normal form if it is in second normal form, and furthermore, the relation's non-key characteristics should not depend on other non-key attributes.

Table No : 1
Table Name : Login
Primary Key : Log_id
Foreign Key : Utype_id
Table Description : To store login data

1 . login

FIELD NAME	**DATATYPE**	**SIZE**	**DESCRIPTION**	**CONSTRAINT**
Logid	int	10	login id	PRIMARY KEY
Username	varchar	50	username	NOT NULL
Password	varchar	50	password	NOT NULL
Status	boolean		active/inactive	NOT NULL
Utype_id	int	11	usertype id	FOREIGN KEY

Table No : 2
Table Name : Patient
Primary Key : P_id
Foreign Key : Log_id, BL_id.
Table Description : To store patients data

2. patient

FIELD NAME	**DATATYPE**	**SIZE**	**DESCRIPTION**	**CONSTRAINT**
P_id	int	10	patient id	PRIMARY KEY
Log_id	int	10	Login id	FOREIGN KEY
P_name	varchar	10	patient name	NOT NULL
P_address	varchar	50	patient address	NOT NULL
P_dob	date		patient date of birth	NOT NULL
P_phone	number	10	patient phone number	NOT NULL
BL_id	int	11	Blood group id	FOREIGN KEY

Table No : 3
Table Name : doctor
Primary Key : D_id
Foreign Key : Log_id, BL_id.
Table Description : To store doctors data

3. doctor

FIELD NAME	**DATATYPE**	**SIZE**	**DESCRIPTION**	**CONSTRAINT**
D_id	int	10	Doctor id	PRIMARY KEY
Log_id	int	10	Login id	FOREIGN KEY
Dept_id	int	10	Department id	FOREIGN KEY
D_name	varchar	50	Doctor name	NOT NULL
D_pos	varchar	50	Doctor position	NOT NULL
D_status	bigint	10	Doctor’s phone	NOT NULL
D_license	varchar	20	Doctor license	NOT NULL
D_specialization_id	int	11	Specialization id	FOREIGN KEY

Table No : 4
Table Name : doctoreducation
Primary Key : D_edu_id
Foreign Key : D_id
Table Description : To store doctor education details

4. doctoreducation

FIELD NAME	**DATATYPE**	**SIZE**	**DESCRIPTION**	**CONSTRAINT**
D_edu_id	int	10	Doctor education id	PRIMARY KEY
D_edu_qualification	varchar	50	Doctors education qualification	NOT NULL
D_id	int	11	Doctor id	FOREIGN KEY

Table No : 5
Table Name : department
Primary Key : Dept_id
Table Description : To store Department details

5. department

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
Dept_id	int	10	Department id	PRIMARY KEY
Dept_name	varchar	50	Department name	NOT NULL
Dept_pic	varchar	50	Department name	NOT NULL
Dept_about	varchar	500	Department name	NOT NULL

Table No : 6
Table Name : specializations
Primary Key : D_specialization_id
Table Description : To store Department specialization details

6. specializations

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
D_specialization_id	int	10	specialization id	PRIMARY KEY
D_specialization	varchar	50	specialization	NOT NULL

Table No : 7
Table Name : schedule
Primary Key : S_id
Foreign Key : W_id , D_id
Table Description : To store Doctor Schedule details

7. schedule

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
S_id	int	10	Schedule id	PRIMARY KEY
S_Time	Time		Starting time	NOT NULL
E_time	Time		Ending time	NOT NULL
W_id	int	10	Week id	FOREIGN KEY
D_id	int	10	Doctor id	FOREIGN KEY

Table No : 8
Table Name : booking
Primary Key :B_id
Foreign Key : P_id , D_id, S_id.
Table Description : To store Appointment Booking details

8. booking

FIELD NAME	**DATATYPE**	**SIZE**	**DESCRIPTION**	**CONSTRAINT**
B_id	int	10	Booking id	PRIMARY KEY
P_id	int	10	Patient id	FOREIGN KEY
D_id	int	10	Doctor id	FOREIGN KEY
S_id	int	10	Schedule id	FOREIGN KEY
B_date	date		Booking date	NOT NULL

Table No : 9
Table Name : usertype
Primary Key : Utype_id
Table Description : To store usertype details

9. usertype

FIELD NAME	**DATATYPE**	**SIZE**	**DESCRIPTION**	**CONSTRAINT**
Utype_id	int	11	usertype id	PRIMARY KEY
Usertype	varchar	10	usertype	NOT NULL

Table No : 10
Table Name : profileimages
Primary Key : Pro_id
Foreign Key : Log_id , Utype_id
Table Description : To store profileimages.

10. profileimages

FIELD NAME	**DATATYPE**	**SIZE**	**DESCRIPTION**	**CONSTRAINT**
Pro_id	int	11	Profile pic id	PRIMARY KEY
Pro_pics	varchar	1000	Profile pics	NOT NULL
Log_id	int	11	Login id	FOREIGN KEY
Utype_id	int	11	Usertype id	FOREIGN KEY

Table No : 11
Table Name : bloodgroup
Primary Key : BL_id
Table Description : To store possible bloodgroups.

11.bloodgroup

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
BL_id	int	11	Bloodgroup id	PRIMARY KEY
BL_group	varchar	50	Blooodgroup	NOT NULL

Table No : 12
Table Name : disease_his
Primary Key : DH_id
Foreign Key : P_id
Table Description : To store Disease history of patients.

12. disease_his

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
DH_id	int	11	Disease history id	PRIMARY KEY
Disease	varchar	50	Disease	NOT NULL
P_id	int	11	Patient id	FOREIGN KEY

Table No : 13
Table Name : family_disease
Primary Key : FD_id
Foreign Key : P_id
Table Description : To store Family Disease history of patients

13.family_disease

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
FD_id	int	11	Family disease id	PRIMARY KEY
FD_disease	varchar	50	Family disease	NOT NULL
P_id	int	11	Patient id	FOREIGN KEY

Table No : 14
Table Name : healthpackage
Primary Key : HP_id
Table Description : To store healthpackage details.

14.healthpackage

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
HP_id	int	11	Health package id	PRIMARY KEY
HP_title	varchar	100	Health package pics	NOT NULL
HP_patientno	int	50	Maximum patients allowed	NOT NULL
HP_price	int	100	Health package price	NOT NULL
HP_from	date		Health package from date	NOT NULL
HP_upto	date		Health package to date	NOT NULL
HP_remark	varchar	500	Health package remarks	NOT NULL
HP_pic	varchar	500	Health package remarks	NOT NULL

Table No : 15
Table Name : hpack_assign
Primary Key : HPA_id
Foreign Key : FD_disease, LB_id
Table Description : To store labtest associated with each health package.

15. hpack_assign

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
HPA_id	int	11	Health package join table id	PRIMARY KEY
FD_disease	varchar	50	Health package id	FOREIGN KEY
LB_id	int	11	Lab id	FOREIGN KEY

Table No : 16
Table Name : hpack_book
Primary Key : HPB_id
Foreign Key : HP_id , P_id
Table Description : To store health package booking.

16. hpack_book

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
HPB_id	int	11	Health package booking id	PRIMARY KEY
HP_id	varchar	50	Health package id	FOREIGN KEY
P_id	int	11	Patient id	FOREIGN KEY
HPB_date	date		Health package booking date	NOT NULL
status	Boolean	1	status	NOT NULL

Table No : 17
Table Name : hpack_report
Primary Key : HR_id
Foreign Key : HPB_id
Table Description : To store health package reports

17. hpack_report

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
HR_id	int	11	Health package report id	PRIMARY KEY
HR_file	varchar	50	Health package report file	NOT NULL
HPB_id	int	11	Health package booking id	FOREIGN KEY
HR_date	datetime		Health package report date	NOT NULL

Table No : 18
Table Name : imr
Primary Key : IMR_id
Foreign Key : SMC_id, D_id
Table Description : To store Doctor imr number and details.

18. imr

FIELD NAME	**DATATYPE**	**SIZE**	**DESCRIPTION**	**CONSTRAINT**
IMR_id	int	11	IMR id	PRIMARY KEY
IMR_number	int	20	IMR number	NOT NULL
SMC_id	int	11	State medical council id	FOREIGN KEY
D_id	int	11	Doctor id	FOREIGN KEY

Table No : 19
Table Name : labrefassign
Primary Key : LA_id
Foreign Key : B_id
Table Description : To store LabTests reffered to patients .

19. labrefassign

FIELD NAME	**DATATYPE**	**SIZE**	**DESCRIPTION**	**CONSTRAINT**
LA_id	int	11	Lab referral join table id	PRIMARY KEY
B_id	int	11	Booking id	FOREIGN KEY
LA_date	datetime		Lab referral join table date	NOT NULL
LA_resultstatus	Boolean	1	Lab referral join table status	NOT NULL

Table No : 20
Table Name : labreferal
Primary Key : LBR_id
Foreign Key : LA_id, LB_id
Table Description : To store LabTests reffered by doctors.

20. labreferal

FIELD NAME	**DATATYPE**	**SIZE**	**DESCRIPTION**	**CONSTRAINT**
LBR_id	int	11	Lab referral id	PRIMARY KEY
LB_id	int	11	Lab test id	FOREIGN KEY
LA_id	int	11	Lab referral join table id	FOREIGN KEY

Table No : 21
Table Name : labresult
Primary Key : LR_id
Foreign Key : LA_id
Table Description : To store lab results.

21. labresult

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
LR_id	int	11	Lab result id	PRIMARY KEY
LR_file	varchar	500	Lab result file	NOT NULL
LA_id	datetime		Lab referral join table id	FOREIGN KEY
LR_date	Boolean	1	Lab result date	NOT NULL

Table No : 22
Table Name : labtests
Primary Key : LB_id
Table Description : To store Available lab tests and details.

22. labtests

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
LB_id	int	11	Lab test id	PRIMARY KEY
LB_test	varchar	100	Lab Test	NOT NULL
LB_price	int	100	Lab Test Price	NOT NULL

Table No : 23
Table Name : medicalhistory
Primary Key : MedHis_id
Foreign Key : P_id
Table Description : To store Medical history of patients.

23. medicalhistory

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
MedHis_id	int	11	Medical history id	PRIMARY KEY
P_id	int	11	Patient id	FOREIGN KEY
MedHis_disease	varchar	50	Medical history disease details	NOT NULL
MedHis_treatment	varchar	50	Medical history treatment	NOT NULL
MedHis_detail	varchar	500	Medical history details	NOT NULL
MedHis_Report	varchar	100	Medical history report file	NOT NULL

Table No : 24
Table Name : medicine
Primary Key : Medi_id
Table Description : To store Available medicine.

24. medicine

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
Medi_id	int	11	Medicine id	PRIMARY KEY
Med_name	varchar	100	Medicine name	NOT NULL
Med_price	int	50	Medicine price	NOT NULL

Table No : 25
Table Name : medrecord
Primary Key : Med_id
Foreign Key : B_id
Table Description : To store medical reports.

25. medrecord

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
Med_id	int	11	Medical record id	PRIMARY KEY
title	varchar	50	Medical record title	NOT NULL
discription	varchar	50	Medical record description	NOT NULL
precuation	varchar	50	precuations	NOT NULL
B_id	int	11	Booking id	FOREIGN KEY
date	datetime		Medical record date	NOT NULL

Table No : 26
Table Name : priscription
Primary Key : med_id
Foreign Key : B_id, Medi_id
Table Description : To store prescription details
26. priscription

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
med_id	int	11	id	PRIMARY KEY
B_id	int	11	Booking id	FOREIGN KEY
medqty	int	10	Medicine quantity	NOT NULL
meddose	varchar	100	Medicine dose	NOT NULL
Medi_id	int	11	Medicine id	FOREIGN KEY
date	datetime		Prescription date	NOT NULL

Table No : 27
Table Name : referal
Primary Key : referal_id
Foreign Key : D_id , Dept_id, B_id
Table Description : To store refferal details.

27. referral

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
referal_id	int	11	Doctor referral id	PRIMARY KEY
Dept_id	int	11	Referred to department id	FOREIGN KEY
D_id	int	11	Referred to Doctor id	FOREIGN KEY
remarks	varchar	50	Referral remark	NOT NULL
B_id	int	11	Booking id	FOREIGN KEY
date	datetime		Reffered date	NOT NULL

Table No : 28
Table Name : social_his
Primary Key : SH_id
Foreign Key : P_id
Table Description : To store Social history details of patients.

28. social_his

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
SH_id	int	11	Social history id	PRIMARY KEY
Tobacco	varchar	50	Tobbaco usage	NOT NULL
Illegal_Drugs	varchar	50	Illegal drugs usage	NOT NULL
Alcohole	varchar	50	Alcohole used	NOT NULL
P_id	int	11	Patient usage	FOREIGN KEY

Table No : 29
Table Name : statemedicalcouncil
Primary Key : SMC_id
Table Description : To store State medical council details.

29. statemedicalcouncil

FIELD NAME	DATATYPE	SIZE	DESCRIPTION	CONSTRAINT
SMC_id	int	11	State medical council id	PRIMARY KEY
SMC_name	varchar	50	State medical council name	NOT NULL

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the process of carefully controlling the execution of software in order to determine whether it behaves as intended. The words verification and validation are frequently used in conjunction with software testing. Validation is the process of examining or evaluating a product, including software, to determine whether it complies with all relevant specifications. One type of verification, software testing, uses methods including reviews, analyses, inspections, and walkthroughs as well. Verifying that what has been specified matches what the user truly desired is the process of validation.

The processes of static analysis and dynamic analysis are additional ones that are frequently related to software testing. Static analysis examines the software's source code, searching for issues and obtaining statistics without actually running the code. Dynamic analysis examines how software behaves while it is running in order to offer data like execution traces, timing profiles, and test coverage details.

Testing is a collection of activities that can be planned ahead of time and carried out in a methodical manner. Testing starts with individual modules and progresses to the integration of the full computer-based system. There are many rules that can be used as testing objectives, and testing is necessary for the system testing objectives to be successful. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a test is successfully carried out in accordance with the aforementioned aims, it will reveal software bugs. Additionally, testing shows that the software functions seem to operate in accordance with the specification and that the performance requirements seem to have been satisfied. There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan suggests a number of required steps that need be taken in order to complete various testing methodologies. The activity that is to be taken is outlined in the test plan. A computer programme, its documentation, and associated data structures are all created by software developers. It is always the responsibility of the software developers to test each of the program's separate components to make sure it fulfils the purpose for which it was intended. In order to solve the inherent issues with allowing the builder evaluate what they have developed, there is an independent test group (ITG). Testing's precise goals should be laid forth in quantifiable language. The test plan should include information on the mean time to failure, cost to find and fix defects, remaining defect density or frequency of occurrence, and test work hours per regression test.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

The smallest unit of software design—the software component or module—is the focus of unit testing, which concentrates verification work. Important control pathways are examined in order to find faults inside the module's border using the component level design description as a guide. The scope set for unit testing and the relative complexity of tests. Unit testing can be carried out simultaneously for numerous components and is white-box focused. Information flow into and out of the programme unit under test is monitored by the modular interface to ensure appropriate operation. To make sure that temporary data is kept in its original form during all phases of an algorithm's execution, the local data structure is inspected. To confirm that each statement in a module has been executed at least once, boundary conditions are evaluated. Finally, each path for managing errors is examined.

Before starting any other test, tests of data flow over a module interface are necessary. All other tests are irrelevant if data cannot enter and depart the system properly. An important duty during the unit test is the selective examination of execution pathways. Error circumstances must be foreseen in good design, and error handling paths must be put up to cleanly reroute or halt work when an error does arise. The final step of unit testing is boundary testing. Software frequently fails at its limits.

In the Sell-Soft System, unit testing was carried out by treating each module as a distinct entity and subjecting them to a variety of test inputs. The internal logic of the modules had some issues, which were fixed. Each module is tested and run separately after coding. All unused code was eliminated, and it was confirmed that every module was functional and produced the desired outcome.

5.2.2 Integration Testing

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interface issues. The goal is to construct a programme structure that has been determined by design using unit tested components. The programme as a whole is tested. Correction is challenging since the size of the overall programme makes it challenging to isolate the causes. As soon as these mistakes are fixed, new ones arise, and the process repeats itself in an apparently unending cycle. All of the modules were integrated after unit testing was completed in the system to check for any interface inconsistencies. A distinctive programme structure also developed when discrepancies in programme structures were eliminated.

5.2.3 Validation Testing or System Testing

The testing process comes to an end here. This involved testing the entire system in its entirety, including all forms, code, modules, and class modules. Popular names for this type of testing include system tests and black box testing.

The functional requirements of the software are the main emphasis of the black box testing approach. To completely exercise all functional requirements for a programme, the software engineer can create sets of input conditions using Black Box testing.

The following sorts of problems are targeted by black box testing: erroneous or missing functions, interface errors, data structure or external data access errors, performance errors, initialization errors, and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system under consideration has its user acceptance assessed; in this case, it must meet the needs of the company. When creating and making modifications as needed, the software should stay in touch with the user and viewpoint system. The following points are considered in this:

Output screen designs, input screen designs, and

The testing mentioned above is carried out using several test data types. In the system testing process, the preparation of test data is crucial. The system under study is evaluated using the test data after it has been prepared. During system testing, faults in the test data are once again found and fixed using the testing procedures described above. The fixes are also logged for use in the future.

Automation Testing

Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it’s a test to double-check that the equipment or software doesexactly what it was designed to do. It tests for bugs, defects, and any other issues that can arisewith product development. Automation testing can be run at any time of the day. It uses scripted sequences to examine the software. It then reports on what’s been found, and this information can be compared with earlier test runs.

Benefits of Automation Testing

Detailed reporting capabilities - Automation testing uses well-crafted test cases for various scenarios. These scripted sequences can be incredibly in-depth, and provide detailed reports that simply wouldn’t be possible when done by a human.

Improved bug detection - One of the main reasons to test a product is to detect bugs and other defects. Automation testing makes this process an easier one. It’s also able to analyzea wider test coverage than humans may be able to.

- Simplifies testing - Testing is a routine part of the operations of most SaaS and tech companies. Making it as simple as possible is key. Using automation is extremely beneficial. When automating test tools, the test scripts can be reused
- Speeds up the testing process - Machines and automated technology work faster than humans. Along with improved accuracy, this is why we use them. In turn, this shortens your software development cycles.

- Reduces human intervention - Tests can be run at any time of day, even overnight, without needing humans to oversee it. Plus, when it's conducted automatically, this can also reduce the risk of human error.

5.2.5 Selenium Testing

Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#, among others. The Selenium testing tool is used to automate tests across browsers for web applications. It's used to ensure high-quality web applications — whether they are responsive, progressive, or regular. Selenium is an open-source tool.

Test cases for a Login Page

Project Name: E-care	
Login Test Case	
Test Case ID: Fun_1	**Test Designed By: K M Abhijith**
Test Priority (Low/Medium/High): High	**Test Designed Date:** 20-07-2022
Module Name: Login Screen	**Test Executed By: Ms. Shelly Shiju George**
Test Title: Verify login with valid username and password	**Test Execution Date:** 20-07-2022
Description: Test the Login Page	
Pre-Condition: User has valid username and password	

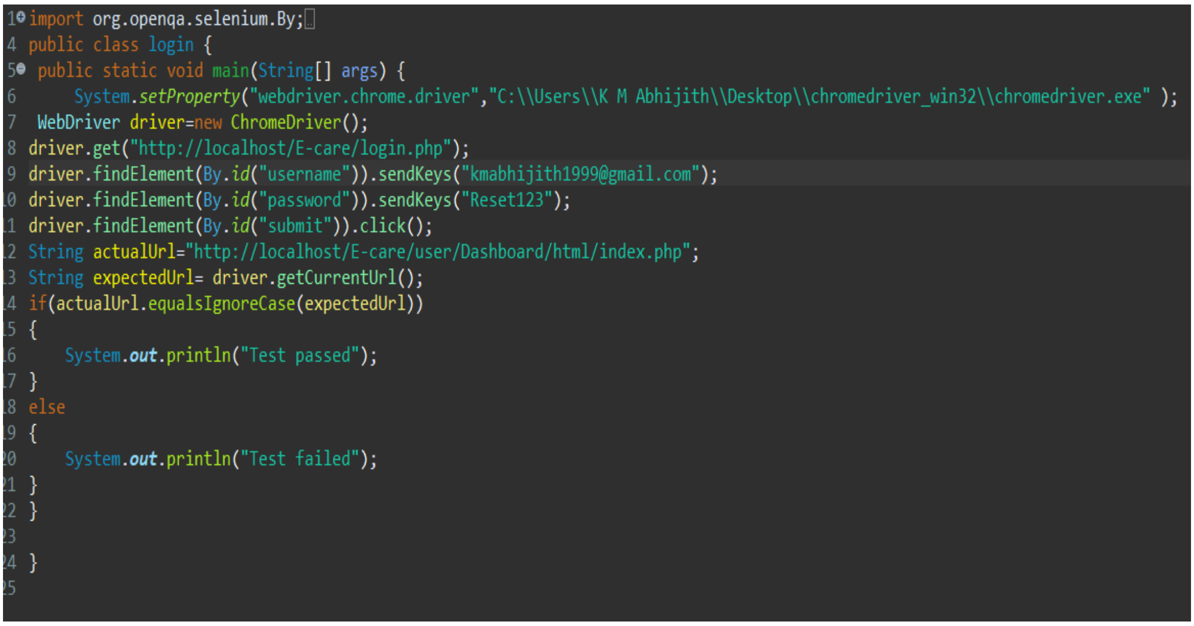
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid username	Username : kmabhijith1999@gmail.com	User should dbe able to Login	User Logged in and navigated to User Dashboard	Pass
3	Provide Valid Password	Password: Reset123			
4	Click on Sign In button				
5	Provide Invalid username or password	Username: kmabhijith@gmail.com Password: User1235	User should not be able to Login	Message for enter valid email id or Password	Pass
6	Provide Null username or Password	Username : null Password: null			
7	Click on Sign In button				

Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database.

Code package

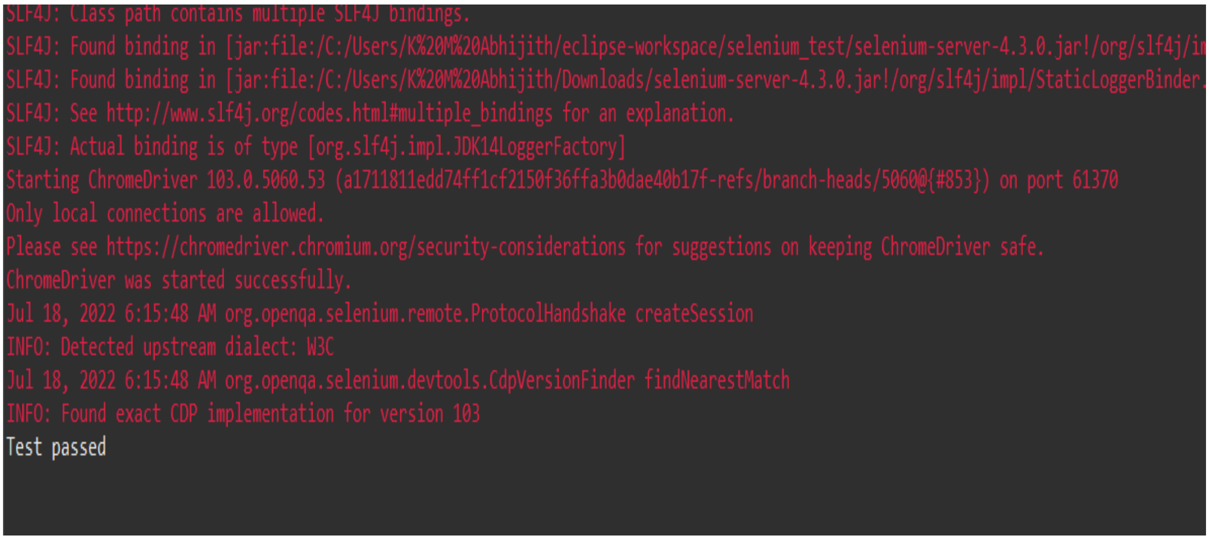
```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class login {
 public static void main(String[] args) {
      System.setProperty("webdriver.chrome.driver","C:\\Users\\K M 
Abhijith\\Desktop\\chromedriver_win32\\chromedriver.exe" );
 WebDriver driver=new ChromeDriver();
driver.get("http://localhost/E-care/login.php");
driver.findElement(By.id("username")).sendKeys("kmabhijith1999@gmail.com");
driver.findElement(By.id("password")).sendKeys("Reset123");
driver.findElement(By.id("submit")).click();
String actualUrl="http://localhost/E-care/user/Dashboard/html/index.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl))
{
     System.out.println("Test passed");
}
else
{
     System.out.println("Test failed");
}
}

     }
```



```
import org.openqa.selenium.By;
public class login {
 public static void main(String[] args) {
     System.setProperty("webdriver.chrome.driver","C:\\Users\\K M Abhijith\\Desktop\\chromedriver_win32\\chromedriver.exe" );
 WebDriver driver=new ChromeDriver();
driver.get("http://localhost/E-care/login.php");
driver.findElement(By.id("username")).sendKeys("kmabhijith1999@gmail.com");
driver.findElement(By.id("password")).sendKeys("Reset123");
driver.findElement(By.id("submit")).click();
String actualUrl="http://localhost/E-care/user/Dashboard/html/index.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl))
{
    System.out.println("Test passed");
}
else
{
    System.out.println("Test failed");
}
}

}
```



```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/Users/K%20M%20Abhijith/eclipse-workspace/selenium_test/selenium-server-4.3.0.jar!/org/slf4j/im
SLF4J: Found binding in [jar:file:/C:/Users/K%20M%20Abhijith/Downloads/selenium-server-4.3.0.jar!/org/slf4j/impl/StaticLoggerBinder.
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.JDK14LoggerFactory]
Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/5060@{#853}) on port 61370
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 18, 2022 6:15:48 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 18, 2022 6:15:48 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed
```

Test cases for Password Update

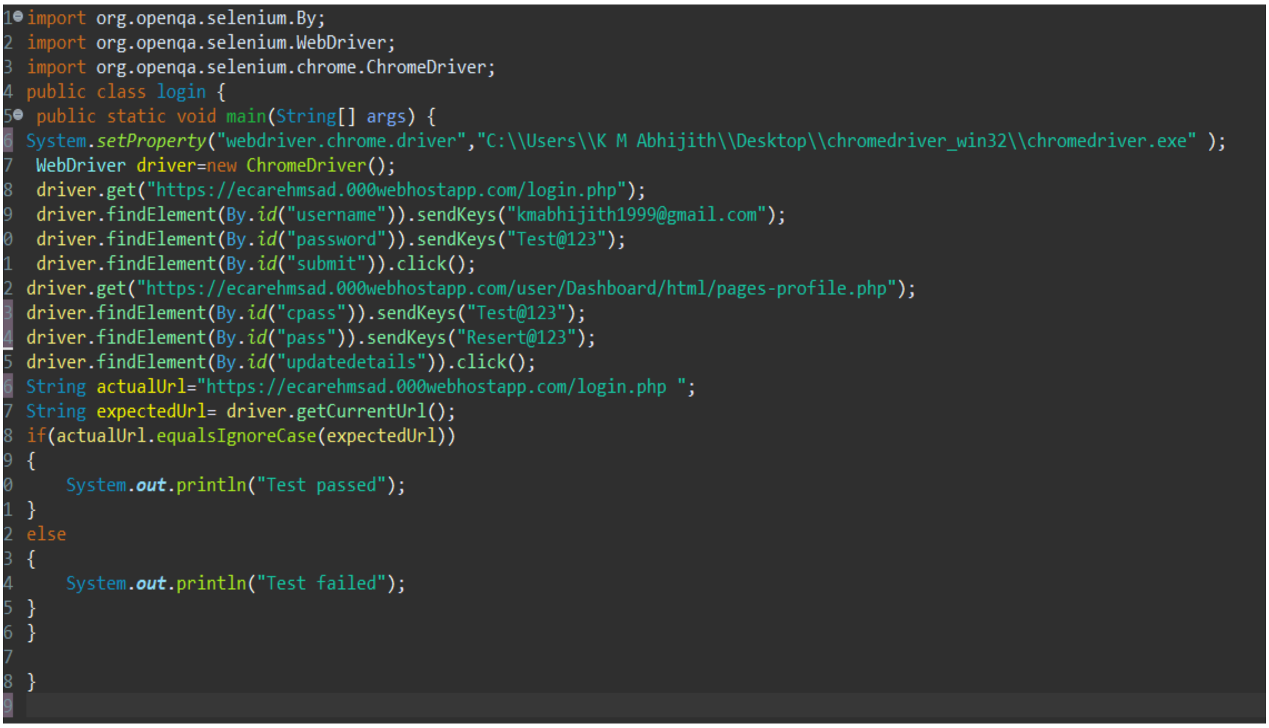
Project Name: E-care					
Update Password Test Case					
Test Case ID: Fun_2			**Test Designed By: K M Abhijith**		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 20-07-2022		
Module Name: Profile Update Screen			**Test Executed By: Ms. Shelly Shiju George**		
Test Title: Update Password by verifying Current password			**Test Execution Date:** 20-07-2022		
Description: Test the Password update page					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Login With Valid credintials		User should be able to login	User Logged in and navigated to User profile	Pass
2	Provide Current Password	Current Password : Test@123	Password Should be updated	User navigated to Login page with Updated password	Pass
3	Provide New password	New Password: Reset@123			
4	Click on Update button				
5	Provide invalid Current password	Current Password : 1234	User should not be able to Update Details	Alert user to enter Valid Password	Pass
6	Provide null Values	Current Password: Null New password :Null			
7	Click on Update				

Post-Condition: User Password is updated in database and can login using updated password.

Code package

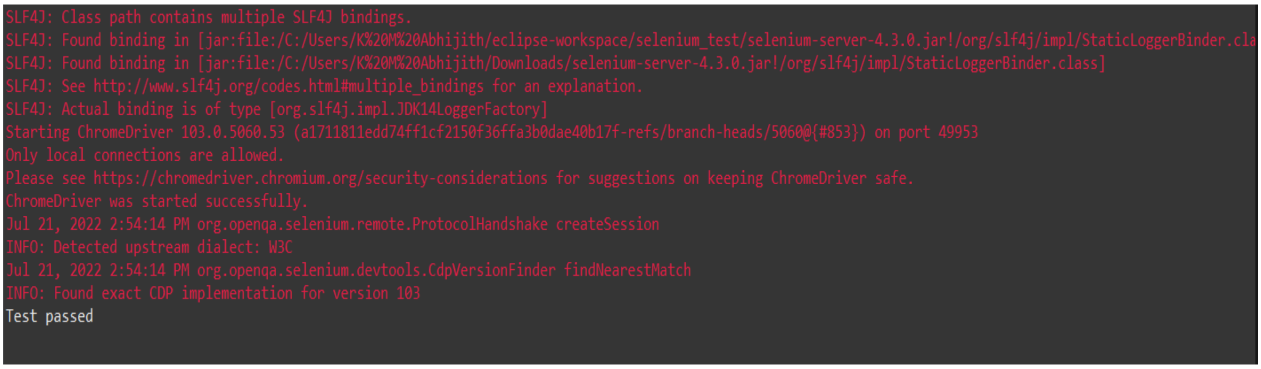
```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class login {
 public static void main(String[] args) {
System.setProperty("webdriver.chrome.driver","C:\\Users\\K M
Abhijith\\Desktop\\chromedriver_win32\\chromedriver.exe" );
 WebDriver driver=new ChromeDriver();
 driver.get("https://ecarehmsad.000webhostapp.com/login.php");
 driver.findElement(By.id("username")).sendKeys("kmabhijith1999@gmail.com");
 driver.findElement(By.id("password")).sendKeys("Test@123");
 driver.findElement(By.id("submit")).click();
driver.get("https://ecarehmsad.000webhostapp.com/user/Dashboard/html/pages-profile.php");
driver.findElement(By.id("cpass")).sendKeys("Test@123");
driver.findElement(By.id("pass")).sendKeys("Resert@123");
driver.findElement(By.id("updatedetails")).click();
String actualUrl="https://ecarehmsad.000webhostapp.com/login.php ";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl))
{
    System.out.println("Test passed");
}
else
{
    System.out.println("Test failed");
}
}

}
```



```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class login {
 public static void main(String[] args) {
System.setProperty("webdriver.chrome.driver","C:\\Users\\K M Abhijith\\Desktop\\chromedriver_win32\\chromedriver.exe" );
 WebDriver driver=new ChromeDriver();
 driver.get("https://ecarehmsad.000webhostapp.com/login.php");
 driver.findElement(By.id("username")).sendKeys("kmabhijith1999@gmail.com");
 driver.findElement(By.id("password")).sendKeys("Test@123");
 driver.findElement(By.id("submit")).click();
driver.get("https://ecarehmsad.000webhostapp.com/user/Dashboard/html/pages-profile.php");
driver.findElement(By.id("cpass")).sendKeys("Test@123");
driver.findElement(By.id("pass")).sendKeys("Resert@123");
driver.findElement(By.id("updatedetails")).click();
String actualUrl="https://ecarehmsad.000webhostapp.com/login.php ";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl))
{
    System.out.println("Test passed");
}
else
{
    System.out.println("Test failed");
}
}

}
```



```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/Users/K%20M%20Abhijith/eclipse-workspace/selenium_test/selenium-server-4.3.0.jar!/org/slf4j/impl/StaticLoggerBinder.cla
SLF4J: Found binding in [jar:file:/C:/Users/K%20M%20Abhijith/Downloads/selenium-server-4.3.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.JDK14LoggerFactory]
Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/5060@{#853}) on port 49953
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 21, 2022 2:54:14 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 21, 2022 2:54:14 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed
```

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. It can be regarded as the most important stage in creating a successful new system since it gives users assurance that the system will operate as intended and be reliable and accurate. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the process of turning a newly revised system design into an operational one, and it simply refers to placing a new system design into operation.

The user department now bears the most of the workload, faces the most disruption, and has the biggest influence on the current system. If the implementation is not well thought out or managed, confusion and mayhem may result.

Implementation encompasses all of the steps used to switch from the old system to the new one. The new system could be entirely different, take the place of an existing manual or automated system, or it could be modified to work better. A reliable system that satisfies organisational needs must be implemented properly. System implementation refers to the process of actually using the built system. This comprises all the processes involved in switching from the old to the new system. Only after extensive testing and if it is determined that the system is operating in accordance with the standards can it be put into use. The system personnel assess the system's viability. The system analysis and design work needed to implement the three key components of education and training, system testing, and changeover will increase in complexity as a system is implemented..

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints. Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended applications. The software development project is frequently commissioned by someone who will not be using it. People first have doubts regarding the software, but we must

Make sure that there isn't an increase in resistance because one must make sure that:The new system's advantages must be known to the active user. Their faith in the software is increased.The user receives the appropriate instruction so that he feels confident using the application.Before examining the system, the user must be aware that the server software needs to be running on the server in order to access the results. The actual process won't happen if the server object is not active and functioning on the server.

6.2.1 User Training

The purpose of user training is to get the user ready to test and modify the system. The people who will be involved must have faith in their ability to contribute to the goal and benefits anticipated from the computer-based system. Training is more necessary as systems get more complicated. The user learns how to enter data, handle error warnings, query the database, call up routines that will generate reports, and execute other important tasks through user training.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

The mystery of system development is maintenance. When a software product is in the maintenance stage of its lifecycle, it is actively working. A system should be properly maintained after it has been effectively implemented. An essential part of the software development life cycle is system maintenance. In order for a system to be flexible to changes in the system environment, maintenance is required. Of course, software maintenance involves much more than just "Finding Mistakes."

6.2.4 Hosting

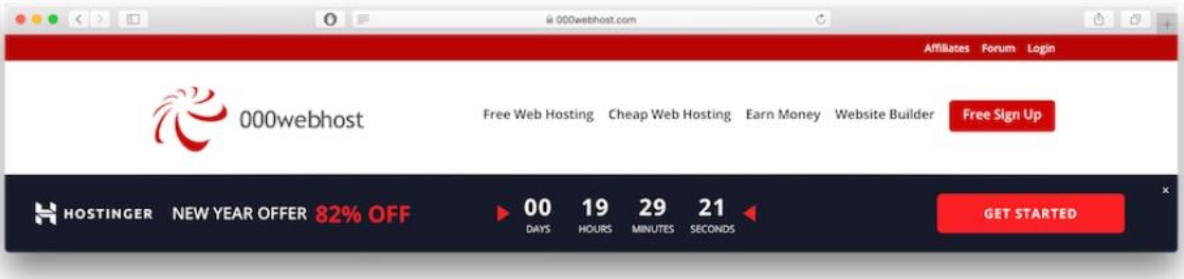
When a facilitating supplier distributes space on a web server for an online site to store its files, they are facilitating an online site. Web facilitating makes the records that include web site (code, pictures, etc.) accessible for seeing online. Every website you've ever gone to is facilitated on a server. The sum of space apportioned on a server to website depends on the sort of facilitating. The most sorts of facilitating are shared, committed, VPS and affiliate. They are separated by the kind of innovation utilized for the server, the level of administration given and the extra administrations on offer.

000Webhost

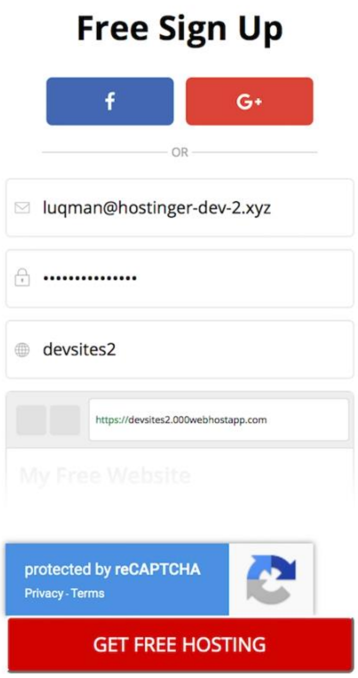
Free web facilitating tends to be so constrained in capabilities and highlights that client must pay to induce what they need. However, 000webhost may be a free website facilitating arrangement that gives an cluster of important highlights, counting web site builder, WordPress back, and no advertisements. Clients can overhaul to a paid arrange to induce indeed more highlights and back, but based on our surveys, 000webhost is the most excellent free web facilitating arrangement for those who are genuinely on a tight budget.

Setting up website in 000webhost

Step 1: Head on to 000webhost.com



Step 2 : Sign up using valid email



Free Sign Up

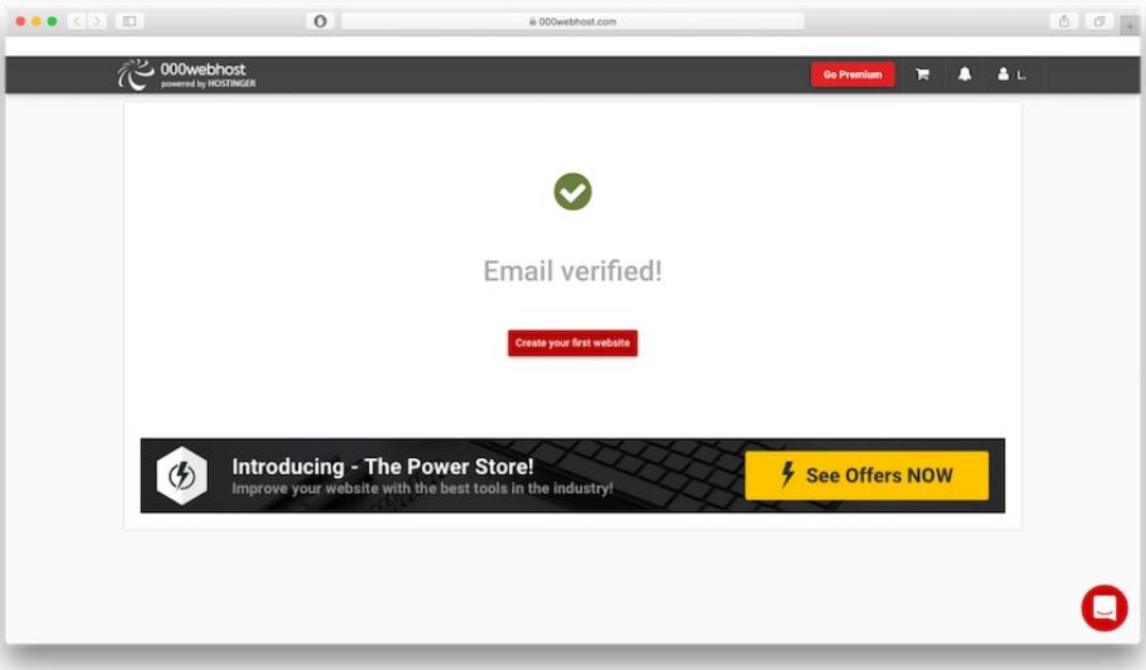
luqman@hostinger-dev-2.xyz

•••••••••••••••

devsites2

https://devsites2.000webhostapp.com

Step 3 : Verify mail id by clicking on the link send by 000webhost.com

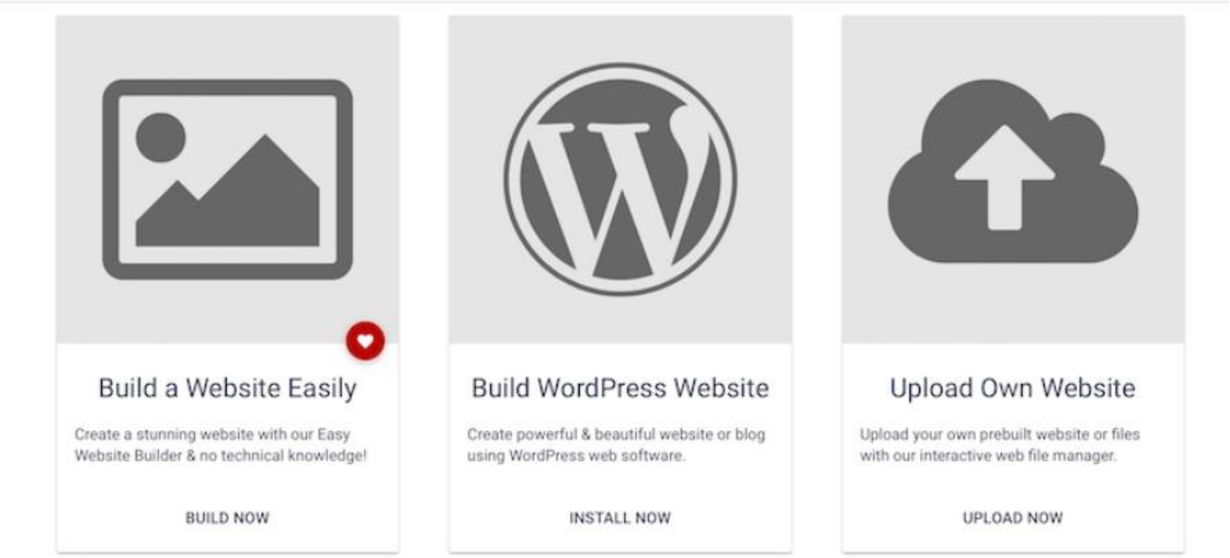


Email verified!

000webhost.com
000webhost
powered by HOSTINGER
Go Premium
Create your first website
Introducing - The Power Store!
Improve your website with the best tools in the industry!
See Offers NOW

Step 4 : search for a domain name and confirm (Provided by 000webhost)

Step 5: Upload website files



Build a Website Easily

Create a stunning website with our Easy Website Builder & no technical knowledge!

BUILD NOW

Build WordPress Website

Create powerful & beautiful website or blog using WordPress web software.

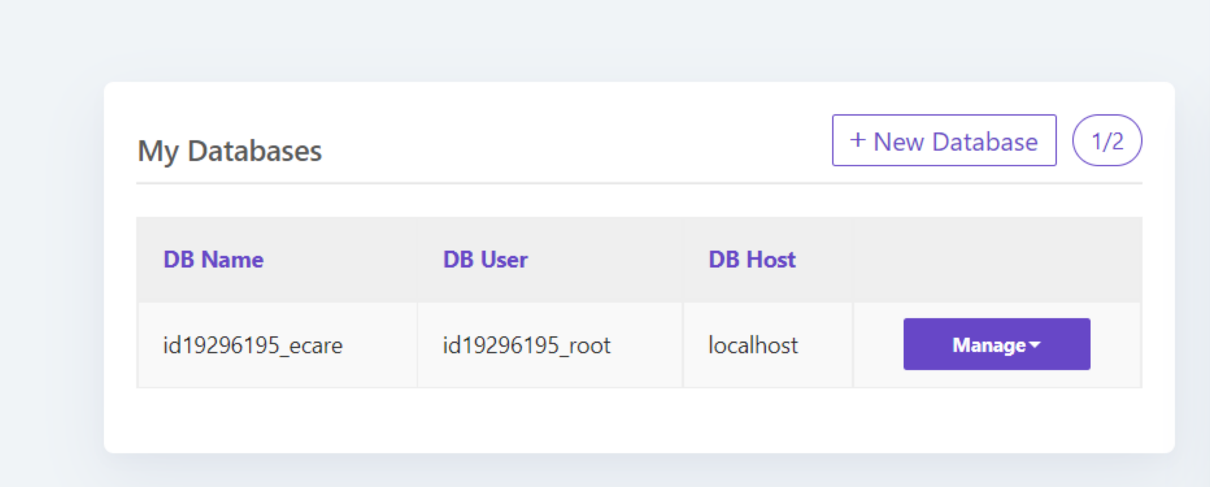
INSTALL NOW

Upload Own Website

Upload your own prebuilt website or files with our interactive web file manager.

UPLOAD NOW

Step 6: Upload database

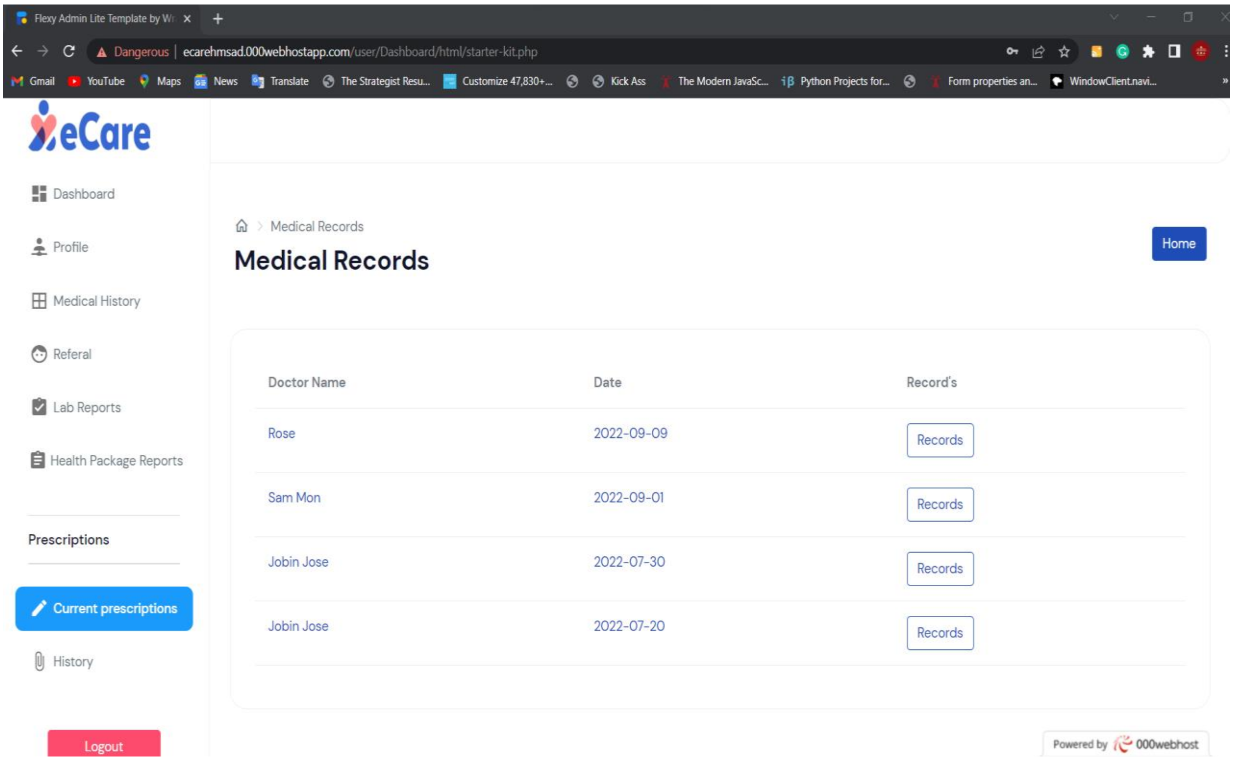


My Databases

DB Name	DB User	DB Host	
id19296195_ecare	id19296195_root	localhost	Manage ▾

Step 7: update connection file with details provided by the system

Step 8 : Hosted website



Medical Records

Medical Records

Doctor Name	Date	Record's
Rose	2022-09-09	Records
Sam Mon	2022-09-01	Records
Jobin Jose	2022-07-30	Records
Jobin Jose	2022-07-20	Records

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for patients to book their slot for consultation by viewing profile of doctors. Provides lots of advantages like search doctors, view profile of doctors, enhanced user interface, view medical reports, lab reports and may more.

7.2 FUTURE SCOPE

- The proposed system is designed in such a way that the payment should be done in online mode.
- Patients can able to add complaints and feedbacks etc.
- Data security can be enhanced.
- Android app for E-care.

18% Overall Similarity

Top sources found in the following databases:

- 12% Internet database
- 7% Publications database
- Crossref database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	G.S. Ajith, M.G. Girija, Jinson Devis. "Poly House Environment Monitori... Crossref	5%
2	issuu.com Internet	3%
3	globalapptesting.com Internet	3%
4	coursehero.com Internet	1%
5	seminarprojects.com Internet	1%
6	Ansem Ben Cheikh, Agnès Front, Jean-Pierre Giraudin, Stéphane Coulo... Crossref	1%
7	docshare.tips Internet	<1%
8	upgrad.com Internet	<1%
9	2019carscomingout.com Internet	<1%

10	ia801708.us.archive.org Internet	<1%
11	WWW.TOOLSQA.COM Internet	<1%
12	utpedia.utp.edu.my Internet	<1%
13	geeksforgeeks.org Internet	<1%
14	readthedocs.web.cern.ch Internet	<1%
15	seminartopicsforcomputerscience.com Internet	<1%